Statistically Reliable and Secure Message Transmission in Directed Networks[☆]

Arpita Patra[*,1,2], Ashish Choudhury[1,3], C. Pandu Rangan[1,4]

Abstract

Consider the following problem: a sender **S** and a receiver **R** are part of a directed synchronous network and connected through intermediate nodes. Specifically, there exists n node disjoint paths, also called as *wires*, which are directed from **S** to **R** and u wires, which are directed from **R** to **S**. Moreover, the wires from **S** to **R** are disjoint from the wires directed from **R** to **S**. There exists a centralized, static adversary $\mathcal{A}_t^{static}$, who has *unbounded computing power* and who can control at most t wires between **S** and **R** in Byzantine fashion. **S** has a message $m^{\mathbf{S}}$, which we wants to send to **R**. The challenge is to design a protocol, such that after interacting in phases [5] as per the protocol, **R** should correctly output $m^{\mathbf{R}} = m^{\mathbf{S}}$, except with error probability $2^{-\Omega(\kappa)}$, where κ is the error parameter. This problem is called as *statistically reliable message transmission* (SRMT). The problem of *statistically secure message transmission* (SSMT) has an additional requirement that at the end of the protocol, $m^{\mathbf{S}}$ should be information theoretically secure from $\mathcal{A}_t^{static}$.

Desmedt et.al [14, 55] have given the necessary and sufficient condition for the existence of SRMT and SSMT protocols in the above settings. They also presented an SSMT protocol, satisfying their characterization. The authors in [14, 55] claimed that their protocol is *efficient* and has polynomial computational and communication complexity. However, we show that it is not so. That is, we specify an adversary strategy, which may cause the protocol to have exponential computational and communication complexity [6]. We then present

[☆]This is an extended, modified and elaborate version of [34]

[*]Corresponding author

Email addresses: arpitapatra_10@yahoo.co.in, arpitapatra10@gmail.com (Arpita Patra), partho_31@yahoo.co.in, partho31@gmail.com (Ashish Choudhury), prangan55@gmail.com, prangan55@yahoo.com (C. Pandu Rangan)

[1]Department of Computer Science and Engineering, IIT Madras, Chennai India 600036.

[2]Financial support from Microsoft Research India acknowledged.

[3]Financial Support from Infosys Technology India Acknowledged.

[4]Work Supported by Project No. CSE/05/06/076/DITX/CPAN on Protocols for Secure Computation and Communication, Sponsored by Department of Information Technology, Govt. of India.

[5]A phase is a send from **S** to **R** or vice-versa.

[6]In fact, this is applicable for the SRMT and SSMT protocols presented in the preliminary version of this paper [34].

new and efficient SRMT and SSMT protocols, satisfying the characterization of [14, 55]. Finally we show that the our proposed protocols are communication optimal by deriving lower bound on the communication complexity of SRMT and SSMT protocols. As far our knowledge is concerned, our protocols are the first communication optimal SRMT and SSMT protocols in directed networks.

Key words: Information Theoretic Security, Error Probability, Optimality, Directed Networks.

---

1. Introduction

Achieving reliable and secure communication is a fundamental problem in the theory of communication. In modern applied network security, there is a lot of emphasis on the use of virtual private networks (using cryptography), firewalls, virus scanners, etc. However, routers too are vulnerable. Two problems have been identified if a router node is hacked. The hacker can shut down the node or forward incorrect information to the adjacent nodes in the network [17, 24]. Hence there is a need for considering an adversary who can disrupt the network in variety of ways. The problem of *reliable message transmission* (RMT) and *secure message transmission* (SMT) perfectly captures the scenario when a specific node in the network intends to send a message to another *non-adjacent* node with the help of other nodes and edges in the network, some of which may be hacked (corrupted) by an adversary.

Let a sender $\mathbf{S}$ and a receiver $\mathbf{R}$ are part of an unreliable connected network, where $\mathbf{S}$ is connected to $\mathbf{R}$ through intermediate nodes. To study the cumulative or combined effect of the faults in the network, we assume the existence of an abstract entity called *centralized adversary*, who can disrupt the communication and computation of some of the intermediate nodes in variety of ways. Moreover, we assume that the adversary has *unbounded computing power*.

In the problem of *reliable message transmission* (RMT), $\mathbf{S}$ has a message $m^{\mathbf{S}}$, which he wants to *reliably* send to $\mathbf{R}$. The goal is to design a protocol, such that after interacting with $\mathbf{S}$ as per the protocol, $\mathbf{R}$ should correctly output m. Moreover, this should happen, even if some of the intermediate nodes are under the control of the centralized adversary. The problem of *secure message transmission* (SMT) has an additional constraint that the adversary should get *no* information about m what so ever, in *information theoretic sense*. Security against such a powerful adversary is called *information theoretic security* or *non-cryptographic security* or *Shannon Security*, which is also the strongest notion of security. Notice that if $\mathbf{S}$ and $\mathbf{R}$ are connected by a direct edge, then RMT and SMT is straight forward: $\mathbf{S}$ simply sends the message to $\mathbf{R}$. Thus the goal of RMT (SMT) protocol is to simulate a direct, virtual, reliable (secure) link between $\mathbf{S}$ and $\mathbf{R}$, who are connected through intermediate nodes, even in the presence of a computationally unbounded centralized adversary.

RMT and SMT are well-motivated problems, for it being one of the fundamental primitives used by all fault-tolerant distributed algorithms like Byzantine

agreement [29, 16, 28, 18, 19, 9, 30], multiparty computation [57, 23, 10, 8, 42, 13, 6, 7] etc. All these popular fault-tolerant distributed algorithms assume that the underlying network is a complete graph. When the graph is not complete, we can simulate the effect of the missing links using RMT/SMT protocols. There is another motivation to study SMT problem. Currently, all existing public key cryptosystems, digital signature schemes are based on the hardness assumptions of certain number theoretic problems. With the advent of new computing paradigms, such as quantum computing [47] and increase in computing speed, may render these assumptions ineffective. Hence it is worthwhile to look for information theoretically secure message transmission schemes.

There are various settings in which RMT and SMT problem has been studied extensively in the past. For example, the underlying network model may be undirected graph [17, 45, 44, 49, 4, 38, 20, 52, 26], directed graph [41, 14, 33, 39, 55, 34, 36, 35, 56] or hypergraph [22, 14, 43, 50]. The communication in the network could be synchronous [17, 45] or asynchronous [44, 48, 12]. The faults could be passive, fail-stop, Byzantine or sometimes mixed/hybrid faults [11]. The number of faulty nodes may be bounded by a fixed constant (threshold adversary) [17, 45, 46] or the potential sets of faulty nodes may be described by a collection of subsets of nodes (non-threshold adversary) [25, 41, 48, 50, 54, 15], while the adversary may be mobile [37, 53, 11] or static [17, 45]. The protocols can be perfect, having no error [17, 26] or may be statistical, having negligible error probability [22, 14, 43, 34, 51]. In general, we may use the following parameters to categorize the different settings in which RMT and SMT problem can be studied:

1. Underlying network;
2. Type of communication;
3. Adversary Capacity;
4. Type of faults; and
5. Type of security.

Irrespective of the settings in which RMT and SMT are studied, the following issues are common:

(i) Possibility: What is the necessary and sufficient condition for the existence of a protocol in a given network?

(ii) Feasibility: Once the existence of a protocol is ensured then does there exist a polynomial time efficient protocol on the given network?

(iii) Optimality: Given a message of specific length, what is the minimum communication complexity (lower bound) needed by any protocol to transmit the message and how to design a protocol whose total communication complexity matches the lower bound on the communication complexity?

In this paper, we study the above issues in the context of *statistical* RMT and SMT in directed synchronous network. We call *statistical* RMT and SMT as SRMT and SSMT respectively. We now define SRMT and SSMT. More formal and rigorous definition will appear in section 2.

1. An RMT protocol is called δ-reliable, for any $\delta = 2^{-\Omega(\kappa)}$, where κ is the error parameter, if at the end of the protocol, $\mathbf{R}$ correctly outputs $\mathbf{S}$'s message, except with error probability δ. Moreover, this should hold, irrespective of the behavior of the adversary.
2. An SMT protocol is called ϵ-secure, for any $\epsilon = 2^{-\Omega(\kappa)}$, where ϵ is the error parameter, if at the end of the protocol, the adversary does not get any any information about $\mathbf{S}$'s message, except with probability ϵ.
3. A message transmission protocol is called (ϵ, δ)-secure, if it is ϵ-secure and δ-reliable.
4. An RMT protocol is called *perfectly reliable*, also called as PRMT, if it is 0-reliable.
5. An RMT protocol is called *statistically reliable*, also called as SRMT, if it is δ-reliable.
6. A message transmission protocol is called *perfectly secure*, also called as PSMT, if it is $(0, 0)$-secure.
7. A message transmission protocol is called *statistically secure*, also called as SSMT, if it is $(0, \delta)$-secure.

1.1. Motivation of Our Work

The PRMT and PSMT problem has been studied extensively over the past three decades in undirected synchronous network model tolerating threshold static Byzantine adversary and the issues related to POSSIBILITY, FEASIBILITY and OPTIMALITY have been completely resolved (see [17, 45, 49, 32, 4, 38, 52, 20, 5, 26]). Also, the issues related to the POSSIBILITY, FEASIBILITY and OPTIMALITY have been completely resolved for SRMT and SSMT in undirected synchronous network model tolerating threshold static Byzantine adversary (see [21, 27, 40]). However, all the above results are in undirected network model, where the communication link between any two nodes in the network is bi-directional. However, in practice not every communication channel may admit bi-directional communication. For instance, a base-station may communicate to even a far-off hand-held device but the other way round communication may not be possible. In such a scenario, it is more appropriate to model the underlying network as a directed graph. Motivated by this, Desmedt et.al [14, 55] introduced the problem of PRMT, PSMT, SRMT and SSMT in directed network and resolved the issue of POSSIBILITY. Recently, [36, 35, 41, 56] resolved the issue related to FEASIBILITY and OPTIMALITY of PRMT and PSMT in synchronous directed networks, tolerating static adversary. However, nothing is known related to the FEASIBILITY and OPTIMALITY of SRMT and SSMT protocols in directed networks. So in this work, we try to completely resolve these issues.

Remark 1 (A Note on the Terminology SRMT and SSMT). *In [34], we have used the terms URMT and USMT for SRMT and SSMT respectively. The reason for the change of terminology in this paper is as follows: In the literature of secure multiparty computation (MPC), protocols with negligible error probability are usually referred as statistical MPC. Since SRMT and SSMT protocols*

will be used as a black box in statistical MPC to simulate a virtual complete network, we prefer to change the terminology from URMT and USMT to SRMT and SSMT respectively.

2. Network Model and Definitions

The underlying network is a connected, synchronous network represented by a directed graph where **S** and **R** are two *non-adjacent* nodes of the graph. All the arcs in the network are reliable and secure but the nodes can be corrupted. The intermediate nodes between **S** and **R** are oblivious, message passing nodes and they do no computation of their own. Their only task is to pass information from their predecessor node to their successor node.

We assume the presence of a static, threshold adversary $\mathcal{A}_t^{static}$, having *unbounded computing power*, who can corrupt any disjoint set of t nodes in the graph (excluding **S** and **R**) in Byzantine fashion. We now formally define Byzantine corruption.

Definition 1 (Byzantine Corruption). *A node P is said to be Byzantine corrupted if the adversary fully control the actions of P. The adversary will have full access to the computation and communication of P and can force P to deviate from the protocol and behave arbitrarily.*

We assume that the adversary is a *centralized* adversary and can collectively pool the data from the nodes under its control and use it according to his own choice in any manner. The adversary is static and corrupts t nodes at the beginning of the protocol. Moreover, a node which is under the control of the adversary will remain so throughout the protocol. Following the approach of [14], we abstract away the network and concentrate on solving SRMT and SSMT problem for a single pair of processors, the *sender* **S** and the *receiver* **R**, connected by node disjoint paths, also called as *wires*, which are directed either from **S** to **R** or vice-versa. More specifically, we assume that there are n wires from **S** to **R**, denoted by $f_1, \ldots, f_n$ and u wires from **R** to **S**, denoted by $b_1, \ldots, b_u$. Moreover, the wires from **S** to **R** are node disjoint from the wires which are directed from **R** to **S**. The n wires from **S** to **R** are also called as *top band*, while the u wires from **R** to **S** are called as *bottom band*. In our protocols, we work with the specific values of n and u. That is, we assume that $0 \leq u \leq t$ and $n = \max(2t - u + 1, t + 1)$. This implies that $n = \Theta(t)$. From [14], these many wires in the top band and bottom band are necessary for any SRMT or SSMT protocol.

The reason for the above abstraction is as follows: suppose some intermediate node between **S** and **R** is under the control of the adversary. Then all the paths between **S** and **R** which passes through that node are also compromised. Hence, all the paths between **S** and **R** passing through that node can be modelled by a single wire between **S** and **R**. In the worst case, the adversary can compromise an entire wire by controlling a single node on the wire. Hence, $\mathcal{A}_t^{static}$ having unbounded computing power can corrupt up to t wires between **S**

and $\mathbf{R}$ (including top and bottom band) in Byzantine fashion. Any protocol in the network operates as a sequence of *phases*, where a phase is a communication from $\mathbf{S}$ to $\mathbf{R}$ or vice-versa.

Throughout this paper, we use $m^{\mathbf{S}}$ to denote the message that $\mathbf{S}$ wishes to send to $\mathbf{R}$. The message is assumed to be a sequence of ℓ elements from the finite field $\mathbb{F}$ with $\ell \geq 1$. Without loss of generality, we assume that $m^{\mathbf{S}}$ is selected uniformly and randomly from $\mathbb{F}$. The size of $\mathbb{F}$ is a function of κ which is the error probability of the SRMT and SSMT protocol. Specifically, $F = GF(2^{\kappa})$ and thus each field element can be represented by $\mathcal{O}(\kappa)$ bits. Moreover, without loss of generality, we assume that $n = \text{poly}(\kappa)$. In our protocols, we assume that the messages sent over any wire are from the right domain. Thus if $\mathbf{S}$ ($\mathbf{R}$) is expecting some message in a specific form and if no message arrives then $\mathbf{S}$ ($\mathbf{R}$) assumes some pre-defined message in the specified form. Thus we separately do not consider the case when no message or syntactically incorrect message is received along a wire.

The *phase complexity* of any SRMT/SSMT protocol is the total number of phases taken by the protocol. The *communication complexity* of any SRMT/SSMT protocol is the total number of field elements communicated by $\mathbf{S}$ and $\mathbf{R}$ in the protocol. The *computational complexity* of any SRMT/SSMT protocol is the total amount of computation done by $\mathbf{S}$ and $\mathbf{R}$ in the protocol. Any SRMT/SSMT protocol is called *efficient* if the phase complexity, computational complexity and communication complexity of the protocol is polynomial in n. Since we measure the size of the message in terms of the number of field elements, we also measure the communication complexity in units of field elements.

Let the message to be transmitted be drawn uniformly and randomly from $\mathbb{F}$. We define the View of a node P_j, at any point of the execution of a protocol Π to be the information that P_j can get from its local input to the protocol (if any), all the messages that P_j had earlier sent or received, the protocol code executed by P_j and random coins of P_j. The View of $\mathcal{A}_t^{static}$ at any point of the execution of Π is defined as all the information that $\mathcal{A}_t^{static}$ can get from the Views of all the nodes corrupted by $\mathcal{A}_t^{static}$ (i.e. all the information that these nodes can commonly compute from their Views). For a message $m^{\mathbf{S}} \in \mathbb{F}$, any t-active threshold adversary characterized by $\mathcal{A}_t^{static}$ and any protocol Π, let $\widehat{\Gamma}(\mathcal{A}_t^{static}, m^{\mathbf{S}}, \Pi)$ denote the probability distribution on the View of the adversary $\mathcal{A}_t^{static}$ at the end of the execution of Π. We now give the following definition

Definition 2 (SSMT). *A protocol Π is said to facilitate statistically secure message transmission (SSMT) between $\mathbf{S}$ and $\mathbf{R}$ if for any message $m^{\mathbf{S}}$ drawn from $\mathbb{F}$ and for every adversary $\mathcal{A}_t^{static}$, the following conditions are satisfied:*

1. *Perfect Secrecy: $\widehat{\Gamma}(\mathcal{A}_t^{static}, m^{\mathbf{S}}, \Pi) = \widehat{\Gamma}(\mathcal{A}_t^{static}, m'^{\mathbf{S}}, \Pi)$. That is, the two distributions are identical irrespective of the message transmitted.*
2. *Statistical Reliability: $\mathbf{R}$ should receive $m^{\mathbf{S}}$ correctly, except with error probability $2^{-\Omega(\kappa)}$, where κ is the error parameter.*

Definition 3 (SRMT). *A protocol* Π *is said to facilitate statistically reliable message transmission (SRMT) between* $\mathbf{S}$ *and* $\mathbf{R}$ *if it satisfies statistical reliability condition of SSMT.*

Definition 4 (Communication Optimal SRMT/SSMT Protocol). *Let* Π *be an* r *(*$r \geq 1$*) phase SRMT (SSMT) protocol which reliably (securely) sends a message* $m^{\mathbf{S}}$ *containing* ℓ *(*$\ell \geq 1$*) field elements by communicating* $O(b)$ *field elements, over a directed network. If the lower bound on the communication complexity of any* r *phase SRMT (SSMT) protocol to send* $m^{\mathbf{S}}$ *over such a network is* $\Omega(b)$ *field elements, then* Π *is said to be a communication optimal SRMT (SSMT) protocol to reliably (securely) send* m*.*

3. Existing Literature and Our Contribution

As mentioned earlier, SRMT and SSMT in directed network was first studied by Desmedt et.al. Specifically, they gave the following characterization:

Theorem 1 ([14, 55]). *Suppose there exists* $u \leq t$ *wires in the bottom band and* n *wires in the top band, such that the wires in the top band are disjoint from the wires in the bottom band. Then any SRMT/SSMT protocol tolerating* $\mathcal{A}_t^{static}$ *is possible iff* $n = \max(2t - u + 1, t + 1)$*.*

In [55], the authors presented an SSMT protocol, satisfying the above characterization. Moreover, the authors claimed that their protocol is efficient (see Theorem 3.4 of [55]). As far our knowledge is concerned, these are the only results for SRMT and SSMT in directed networks.

3.1. Our Contributions

We now summarize the contributions of this article:

1. We first show that the SSMT protocol presented in [14, 55] is inefficient. We do this by specifying an adversary strategy, which may cause the protocol to have exponential computational and communication complexity. In fact, we show that the same adversary strategy is applicable for the SRMT and SSMT protocols presented in the preliminary version of this paper [34], thus making the SRMT and SSMT protocols of [34] to have exponential computational and communication complexity.
2. We then present new and efficient SRMT and SSMT protocols. These protocols are achieved by making several modifications to the SRMT and SSMT protocols presented in the preliminary version of this paper [34]. In fact, these modifications when applied to the SSMT protocol of [14, 55] will make it efficient. In short, our SRMT and SSMT protocols have the following properties:
   (a) Our SRMT protocol takes $\mathcal{O}(u)$ phases and reliably sends a message containing $\Theta(n^3\kappa)$ bits by overall communicating $\mathcal{O}(n^3\kappa)$ bits. Thus, our SRMT protocol achieves reliability with constant factor overhead.

(b) Our SSMT protocol takes $\mathcal{O}(u)$ phases and has a communication complexity of $\mathcal{O}(n^3\kappa)$ bits. If the entire bottom band is corrupted, then the protocol securely sends a message containing $\Theta(n^2u\kappa)$ bits. Otherwise, the protocol securely sends a message containing $\Theta(n^2\kappa)$ bits.

3. Finally, we show that our SRMT and SSMT protocols are asymptotically communication optimal. For this, we derive the lower bound on the communication complexity of SRMT and SSMT protocols. Specifically, we show the following:
   (a) Any SRMT protocol must communicate $\Omega(\ell\kappa)$ bits to reliably send a a message containing $\ell\kappa$ bits.
   (b) If the entire bottom band is corrupted then any SSMT protocol must communicate $\Omega(\frac{n\ell}{u}\kappa)$ bits to securely send a message containing $\ell\kappa$ bits. On the other hand, if the entire bottom band is not corrupted then any SSMT protocol must communicate $\Omega(n\ell\kappa)$ bits to securely send a message containing $\ell\kappa$ bits.

3.2. Overview of Our Protocols

We first design a three phase SSMT protocol called 3-SSMT, which sends a message containing $\Theta(\frac{n}{3})$ field elements by communicating $\mathcal{O}(n^3)$ field elements. Then using this protocol as a black-box, we design a six phase protocol called 6-Pad, which securely establishes a random, one time pad between **S** and **R**. Then using protocol 6-Pad as a black-box, we design our communication optimal SRMT protocol called SRMT-Optimal, which takes $\mathcal{O}(u)$ phases. Finally, using protocol 6-Pad and protocol SRMT-Optimal as a black-box, we design our $\mathcal{O}(u)$ phase communication optimal SSMT protocol called SSMT-Optimal. The idea behind SSMT-Optimal is as follows: we first execute protocol 6-Pad to securely establish a one time pad between **S** and **R**. Then using this pad, **S** masks the secret message and reliably sends the masked message by using the protocol SRMT-Optimal. After receiving the masked message, **R** can unmask the message by using the pad.

3.3. Comparison of Our Protocols with the Protocols of [54, 46, 51]

The characterization given in Theorem 1 and the protocols proposed by us in this paper considers "wired" network model, where it is assumed that the intermediary nodes are just message forwarding nodes. Under this assumption, we abstract the network in the form of directed wires, directed either from **S** to **R** or **R** to **S**. However, in an *arbitrary directed* network, if the intermediate nodes (other than **S** and **R**) are allowed to carry out computation and communication (beyond just acting as a message forwarding node), as in the case of a *virtual private network (VPN)*, then the wired abstraction results in loss of generality. The insufficiency of wired abstraction in such a network model is pointed out in [54, 46, 51] where characterizations for SRMT/SSMT over the arbitrary network, treating entire graph in its full form are also reported. While authors in [54, 51] have considered threshold adversary, the authors of [46] have

considered non-threshold [5] adversary for their characterization. However, it is likely to take exponential time to verify whether a given arbitrary directed network satisfies the characterization given in [54, 46, 51] for the possibility of SRMT/SSMT. Moreover, the protocols given in [54, 46, 51] require exponential computational and communication complexity and are highly non-intuitive.

It should be noted that abstracting the underlying network to a bunch of wires is *incomparable* to treating the network in its full form. Both are sensible and practical. We need to decide which model to follow based on the characteristic of given underlying network. In this paper, we consider the wire model, as it is relatively simple. Moreover, it is relatively easy to design protocols in wire model, in comparison to design protocols by considering the graph in its entirety.

3.4. The Roadmap

The paper is organized as follows: in the next section, we present the tools that are used in our protocols. In Section 5, we present our three phase SSMT protocol called 3-SSMT. In Section 5.4, we show the inefficiency of the SRMT and SSMT protocols of [14, 34, 55]. Section 6 presents our six phase pad establishment protocol 6-Pad. Our communication optimal SRMT protocol is presented in Section 7. In Section 8, we present our communication optimal SSMT protocol. The lower bounds on the communication complexity of SRMT and SSMT protocols are presented in Section 9. The paper ends with a conclusion and directions for further research.

4. Tools Used in Our Protocols

We now present the tools which are used in this paper.

Definition 5 (Unconditionally Reliable Authentication [42, 14, 21]). *It is used to send a message M over a wire such that if the wire is uncorrupted, then the receiver correctly gets M. On the other hand, if the wire is corrupted, then the receiver does not get M but is able to detect the corruption with very high probability. Though there are several implementations for this well known primitive, we use the following implementation in this paper: Let a random, non-zero $(a, b) \in \mathbb{F}^2$ be securely established between the sender and the receiver in advance. The sender computes $x = URauth(M; a, b) = aM + b$ and sends (M, x) to the receiver over the wire. Let the receiver receive (M', x') along the wire. Receiver verifies $x' \stackrel{?}{=} URauth(M'; a, b)$. If the test fails then the receiver concludes that $M' \neq M$, otherwise $M' = M$. The tuple (a, b) is called authentication key. The probability that $M' \neq M$, but still the receiver fails to detect it is at most $\frac{1}{|\mathbb{F}|}$, which is negligible in our context. Note that a remains information theoretically secure, even if the adversary knows (M, x) by listening the wire.*

[5] A non-threshold adversary is a generalized form of the threshold adversary.

Definition 6 (Unconditionally Secure Authentication [42, 14, 21]). *The goal here is similar to unconditionally reliable authentication. However, we now require an additional requirement that M should be information theoretically secure, even if the wire is under the control of the adversary. Again there are several implementations for this well known primitive. In this paper, we use the following implementation: Let a random, non-zero $(a, b, c) \in \mathbb{F}^3 - \{(0, 0, 0)\}$ be securely established between the sender and the receive in advance. The sender computes $(x, y) = USauth(M; a, b, c) = (M + a, b(M + a) + c)$ and sends (x, y) to the receiver over the wire. Let the receiver receive (x', y') along the wire. The receiver verifies $y' \stackrel{?}{=} bx' + c$. If the test fails then the receiver concludes that the wire is corrupted, else the receiver recovers $x' - a$. It is easy to see that even if the adversary knows (x, y), then also M is information theoretically secure. Moreover, if $(x', y') \neq (x, y)$, then except with error probability $\frac{1}{|\mathbb{F}|}$ (which is negligible), the receiver will be able to detect it.*

Definition 7 (Unconditional Hashing [6]). *Let $(v_1, v_2, \ldots, v_\ell)$ be a random vector from $\mathbb{F}^\ell$, where $\ell > 1$ and $k \in \mathbb{F} - \{0\}$. Then we define $hash(k; v_1, v_2, \ldots, v_\ell) = v_1 + v_2 k + v_3 k^2 + \ldots + v_\ell k^{\ell-1}$. Here k is called the hash key. The probability that two different vectors map to the same hash value for a uniformly chosen hash key is at most $\frac{\ell}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$. If the adversary knows only k and $hash(k; v_1, v_2, \ldots, v_\ell)$, then $\ell - 1$ elements in the vector will be information theoretically secure.*

Definition 8 (Extracting Randomness [49]). *Suppose $\mathbf{S}$ and $\mathbf{R}$ by some means agree on a sequence of n random elements $x = [x_1\ x_2\ \ldots\ x_n] \in \mathbb{F}^n$ such that $\mathcal{A}_t^{static}$ knows $n - f$ components of x, but has no information about the other f components of x. However $\mathbf{S}$ and $\mathbf{R}$ does not know which values are known to $\mathcal{A}_t^{static}$. The goal of $\mathbf{S}$ and $\mathbf{R}$ is to agree on a sequence of f elements $[y_1\ y_2\ \ldots\ y_f] \in \mathbb{F}^f$, such that $\mathcal{A}_t^{static}$ has no information about $[y_1\ y_2\ \ldots\ y_f]$. This is done as follows:*

Algorithm EXTRAND$_{n,f}(x)$: *Let V be a $n \times f$ Vandermonde matrix with elements in $\mathbb{F}$. This matrix is published as a part of the algorithm specification. $\mathbf{S}$ and $\mathbf{R}$ both locally compute the product $[y_1\ y_2\ \ldots\ y_f] = [x_1\ x_2\ \ldots\ x_n]V$.*

5. A Three Phase SSMT Protocol

We now present a three phase SSMT protocol called 3-SSMT. The protocol securely sends a message $m^{\mathbf{S}}$ containing $\frac{n}{3}$ field elements by communicating $\mathcal{O}(n^3)$ field elements. The protocol will be later used in our communication optimal SRMT and SSMT protocol. The protocol uses certain ideas from the SSMT protocol of [14]. In addition, the protocol also uses certain new ideas proposed by us. Before presenting protocol 3-SSMT, we present another three phase SSMT protocol called 3-SSMT-Exponential, which requires exponential

communication and computational complexity. The main reason for presenting 3-SSMT-Exponential is to present the underlying principle, which we try to simulate in protocol 3-SSMT, while maintaining polynomial computation and communication complexity. A protocol some what similar to protocol 3-SSMT-Exponential is also presented in [14]. So in the next section, we first present protocol 3-SSMT-Exponential.

5.1. A Three Phase Exponential SSMT Protocol

Let $\mathcal{P}_1, \ldots, \mathcal{P}_k$ denote the enumeration of all possible $t+1$-sized subset of the wire set $\{f_1, \ldots, f_n, b_1, \ldots, b_u\}$. Thus $k = \binom{2t+1}{t+1}$. Since there are at least $t+1$ honest wires including top and bottom band, there exists at least one path set, say $\mathcal{P}_i$, where $\mathcal{P}_i$ contains all $t+1$ honest wires. Moreover, each path set $\mathcal{P}_j$ will have at least one wire from top band, as there can be at most t wires in the bottom band. If some how $\mathbf{S}$ and $\mathbf{R}$ comes to know the identity of the honest path set $\mathcal{P}_i$, which consists of only honest wires, then $\mathbf{S}$ and $\mathbf{R}$ can share an n-tuple over each wire in $\mathcal{P}_i$. Then adding all such n-tuples, $\mathbf{S}$ and $\mathbf{R}$ can agree on an n-tuple, about which adversary will have no information. Finally, using the elements of the resultant n-tuple as encryption and authentication keys, $\mathbf{S}$ can reliably and securely send $m^{\mathbf{S}}$ over all the wires in the top band, which are present in $\mathcal{P}_i$.

Since, neither $\mathbf{S}$ nor $\mathbf{R}$ will know the exact identity of honest path set $\mathcal{P}_i$ in advance, they have to parallely do the above procedure for all paths sets $\mathcal{P}_1, \ldots, \mathcal{P}_k$. During this process, if the adversary tries to change the information over the wires in any path set then with very high probability, $\mathbf{R}$ will detect this and will neglect the information which is exchanged over the wires in that path set. The protocol is formally given in Fig. 1.

We now prove the properties of protocol 3-SSMT-Exponential.

Lemma 1. *In protocol 3-SSMT-Exponential, adversary will have no information about* $m^{\mathbf{S}}$.

PROOF: Every path set $\mathcal{P}_m$ will have at least one honest wire, either in the top band or in the bottom band. So the adversary will have no information about the n-tuple which is exchanged between $\mathbf{S}$ and $\mathbf{R}$ over that wire. As a result, the adversary will have no information about the keys used by $\mathbf{S}$, corresponding to the path set $\mathcal{P}_m$. The rest now follows from the properties of $USauth$. □

Lemma 2. *In protocol 3-SSMT-Exponential,* $\mathbf{R}$ *will always terminate.*

PROOF: The proof follows from the fact that there exists at least one path set $\mathcal{P}_m$, which will contain only honest wires. □

Lemma 3. *In protocol 3-SSMT-Exponential, if* $\mathbf{R}$ *outputs* $m^{\mathbf{R}}$ *then with very high probability,* $m^{\mathbf{R}} = m^{\mathbf{S}}$.

Figure 1: Protocol 3-SSMT-Exponential: A Three Phase Exponential SSMT Protocol

Phase I: S to R: $\mathbf{S}$ parallely does the following computation and communication corresponding to each path set $\mathcal{P}_m$, for $m = 1, \ldots, k$:

1. Corresponding to each $f_i \in \mathcal{P}_m$, $\mathbf{S}$ selects a random non-zero n-tuple $(x^{\mathbf{S}}_{1,i,m}, \ldots, x^{\mathbf{S}}_{n,i,m})$.
2. $\mathbf{S}$ sends $(x^{\mathbf{S}}_{1,i,m}, \ldots, x^{\mathbf{S}}_{n,i,m})$ to $\mathbf{R}$ over wire f_i.

Phase II: R to S: $\mathbf{R}$ parallely does the following computation and communication, corresponding to each path set $\mathcal{P}_m$, for $m = 1, \ldots, k$:

1. Let $\mathbf{R}$ receive non-zero n-tuple $(x^{\mathbf{R}}_{1,i,m}, \ldots, x^{\mathbf{R}}_{n,i,m})$ from $\mathbf{S}$, over wire f_i, corresponding to each $f_i \in \mathcal{P}_m$.
2. Corresponding to each $b_i \in \mathcal{P}_m$, $\mathbf{R}$ selects a random non-zero n-tuple $(y^{\mathbf{R}}_{1,i,m}, \ldots, y^{\mathbf{R}}_{n,i,m})$.
3. $\mathbf{R}$ sends $(y^{\mathbf{R}}_{1,i,m}, \ldots, y^{\mathbf{R}}_{n,i,m})$ to $\mathbf{S}$ over wire b_i.

Phase III: S to R: $\mathbf{S}$ parallely does the following computation and communication corresponding to each path set $\mathcal{P}_m$, for $m = 1, \ldots, k$:

1. Let $\mathbf{S}$ receive non-zero n-tuple $(y^{\mathbf{S}}_{1,i,m}, \ldots, y^{\mathbf{S}}_{n,i,m})$ from $\mathbf{R}$, over wire b_i, corresponding to each $b_i \in \mathcal{P}_m$.
2. For $i = 1, \ldots, n$, $\mathbf{S}$ computes his version of n keys $\mathcal{C}^{\mathbf{S}}_{i,m} = \sum_{f_j \in \mathcal{P}_m} x^{\mathbf{S}}_{i,j,m} + \sum_{b_j \in \mathcal{P}_m} y^{\mathbf{S}}_{i,j,m}$.
3. For each element of $m^{\mathbf{S}}$ (recall that $|m^{\mathbf{S}}| = \frac{n}{3}$), $\mathbf{S}$ takes three elements from the keys computed in the previous step and computes the set $\mathcal{S}^{\mathbf{S}}_m = \{(c^{\mathbf{S}}_{i,m}, d^{\mathbf{S}}_{i,m}) : i = 1, \ldots, \frac{n}{3}\}$ where $(c^{\mathbf{S}}_{i,m}, d^{\mathbf{S}}_{i,m}) = USauth(m^{\mathbf{S}}_i; \mathcal{C}^{\mathbf{S}}_{3i-2,m}, \mathcal{C}^{\mathbf{S}}_{3i-1,m}, \mathcal{C}^{\mathbf{S}}_{3i,m})$, for $i = 1, \ldots, \frac{n}{3}$.
4. $\mathbf{S}$ sends the set $\mathcal{S}^{\mathbf{S}}_m$ to $\mathbf{R}$ over all the top band wires in the set $\mathcal{P}_m$ and terminates the protocol.

Message Recovery by R: If $\mathbf{R}$ receives $\mathcal{S}^{\mathbf{R}}_m = \{(c^{\mathbf{R}}_{i,m}, d^{\mathbf{R}}_{i,m}) : i = 1, \ldots, \frac{n}{3}\}$ over all f_j's in path set $\mathcal{P}_m$, then corresponding to path set $\mathcal{P}_m$, $\mathbf{R}$ does the following computation:

1. For $i = 1, \ldots, n$, $\mathbf{R}$ computes his version of n keys $\mathcal{C}^{\mathbf{R}}_{i,m} = \sum_{f_j \in \mathcal{P}_m} x^{\mathbf{R}}_{i,j,m} + \sum_{b_j \in \mathcal{P}_m} y^{\mathbf{R}}_{i,j,m}$.
2. For $i = 1, \ldots, \frac{n}{3}$, $\mathbf{R}$ checks whether $d^{\mathbf{R}}_{i,m} \stackrel{?}{=} \mathcal{C}^{\mathbf{R}}_{3i-1,m} c^{\mathbf{R}}_{i,m} + \mathcal{C}^{\mathbf{R}}_{3i,m}$.
3. If the above test passes for all $i = 1, \ldots, \frac{n}{3}$, then $\mathbf{R}$ computes $m^{\mathbf{R}}_i = c^{\mathbf{R}}_{i,m} - \mathcal{C}^{\mathbf{R}}_{3i-2,m}$. $\mathbf{R}$ then concatenates $m^{\mathbf{R}}_1, \ldots, m^{\mathbf{R}}_{\frac{n}{3}}$ to recover $m^{\mathbf{R}}$ and terminates the protocol.

PROOF: Suppose $\mathbf{R}$ outputs $m^{\mathbf{R}}$ from $\mathcal{S}_m^{\mathbf{R}}$, corresponding to path set $\mathcal{P}_m$. Since there is at least one honest wire in $\mathcal{P}_m$ and the adversary has no information about the n-tuple which is exchanged between $\mathbf{S}$ and $\mathbf{R}$ over that wire, it implies that the adversary has no information about the keys computed by $\mathbf{S}$ and $\mathbf{R}$, corresponding to path set $\mathcal{P}_m$. If the path set $\mathcal{P}_m$ is completely honest or if the adversary is passively controlling the wires under its control in $\mathcal{P}_m$, then $\mathbf{S}$ and $\mathbf{R}$ will have the same keys and hence $m^{\mathbf{R}} = m^{\mathbf{S}}$. On the other hand, if the adversary has modified the tuples which are exchanged over the wires under its control in $\mathcal{P}_m$, then $\mathbf{S}$ and $\mathbf{R}$ will have different keys. But still, as explained above, the adversary will have no information about the keys. So from the properties of $USauth$, the adversary can make the verification process successful at $\mathbf{R}$'s end with very negligible probability. Thus $m^{\mathbf{R}} = m^{\mathbf{S}}$ with very high probability. □

Lemma 4. *Protocol 3-SSMT-Exponential requires exponential computation and communication complexity.*

We now proceed to the discussion of protocol 3-SSMT, which is an efficient three phase SSMT protocol. The principle used in 3-SSMT is similar to protocol 3-SSMT-Exponential. However, instead of working with all possible $t+1$-sized subset of wires, $\mathbf{S}$ and $\mathbf{R}$ uses certain mechanism, which allows them to work with at most u path sets, thus making the communication and computation complexity of the protocol polynomial.

5.2. Protocol 3-SSMT: A Three Phase Efficient SSMT Protocol

In protocol 3-SSMT, $\mathbf{R}$ first tries to find whether there exists $t+1$ honest wires in the top band. In order to facilitate $\mathbf{R}$ to do so, $\mathbf{S}$ tries to send $m^{\mathbf{S}}$ using two different methods. If there exists $t+1$ honest wires in the top band then the first method would be successful. However, if there are less that $t+1$ honest wires in the top band, then with very high probability $\mathbf{R}$ will detect this and will conclude that at least one honest wire is present in the bottom band. So $\mathbf{S}$ and $\mathbf{R}$ interacts for two more phases and $\mathbf{S}$ tries to again send $m^{\mathbf{S}}$ using the second method in the third phase. The second method tries to follow the principle used in protocol 3-SSMT, however ensuring that the communication and computation complexity is polynomial in n.

We now begin with the description of protocol 3-SSMT, phase by phase. However, instead of describing the entire protocol in a single shot, we prefer to discuss each phase individually. This would help the reader to understand the nuances and ideas used in each phase. So we begin with the description of first phase of protocol 3-SSMT, which is given in the next subsection.

5.2.1. Phase I of Protocol 3-SSMT

During the first phase of the protocol, $\mathbf{S}$ tries to send $m^{\mathbf{S}}$ using the first method. $\mathbf{S}$ also sends some additional information, which might be useful during second phase, if at all it is executed. **Phase I** of protocol 3-SSMT is formally given in Fig. 2.

We now prove the properties of **Phase I**.

Figure 2: Phase I of Protocol 3-SSMT

Phase I: S to R: **S** does the following computation and communication:

1. **S** selects a random polynomial $M^{\mathbf{S}}(x)$ over $\mathbb{F}$ of degree $n-1+t$ such that the lower order $\frac{n}{3}$ coefficients of $M^{\mathbf{S}}(x)$ are elements of $m^{\mathbf{S}}$.
2. **S** computes $M^{\mathbf{S}}(1), \ldots, M^{\mathbf{S}}(n+t)$.
3. **S** then selects $n+t$ random polynomials $f_1^{\mathbf{S}}(x), \ldots, f_{n+t}^{\mathbf{S}}(x)$ over $\mathbb{F}$, each of degree t, such that for $i = 1, \ldots, n+t$, $f_i^{\mathbf{S}}(0) = M^{\mathbf{S}}(i)$.
4. **S** evaluates each $f_i^{\mathbf{S}}(x)$ at $x = 1, \ldots, n$ to form an n-tuple $f_i^{\mathbf{S}} = [f_i^{\mathbf{S}}(1), \ldots, f_i^{\mathbf{S}}(n)]$.
5. **S** constructs an $(n) \times (n+t)$ matrix T where i^{th} column of T contains the n-tuple $f_i^{\mathbf{S}}$, for $i = 1, \ldots, n+t$. The matrix T is pictorially shown in Table 1. Let $F_i^{\mathbf{S}} = [f_1^{\mathbf{S}}(i), \ldots, f_{n+t}^{\mathbf{S}}(i)]$ denote the i^{th} row of T, for $i = 1, \ldots, n$.
6. For $i = 1, \ldots, n$, **S** sends the following to **R** along wire f_i:
   (a) The vector $F_i^{\mathbf{S}}$;
   (b) A random non-zero *hash key* $\alpha_i^{\mathbf{S}}$ and
   (c) The n-tuple $[v_{1i}^{\mathbf{S}}, \ldots, v_{ni}^{\mathbf{S}}]$, where for $j = 1, \ldots, n$, $v_{ji}^{\mathbf{S}} = hash(\alpha_i^{\mathbf{S}}; F_j^{\mathbf{S}})$.
7. In addition to all above computation and communication, **S** also selects a random non-zero $(n+1)$-tuple $(x_{1,i}^{\mathbf{S}}, \ldots, x_{n+1,i}^{\mathbf{S}})$, which is independent of $F_i^{\mathbf{S}}$, corresponding to every wire f_i, for $i = 1, \ldots, n$. **S** then sends $(x_{1,i}^{\mathbf{S}}, \ldots, x_{n+1,i}^{\mathbf{S}})$ to **R** over wire f_i.

Computation by R at the end of Phase I:

1. Let **R** receive the following over wire f_i, for $i = 1, \ldots, n$:
   (a) The vector $F_i^{\mathbf{R}}$;
   (b) The *hash key* $\alpha_i^{\mathbf{R}}$;
   (c) The n tuple $[v_{1i}^{\mathbf{R}}, \ldots, v_{ni}^{\mathbf{R}}]$ and
   (d) The $(n+1)$-tuple $(x_{1,i}^{\mathbf{R}}, \ldots, x_{n+1,i}^{\mathbf{R}})$.
2. For $i = 1, \ldots, n$, **R** computes $Support_i = |\{f_j : v_{ij}^{\mathbf{R}} = hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}})\}|$. If $Support_i \geq t+1$, then **R** concludes that $F_i^{\mathbf{R}}$ is a valid row of T. Otherwise, **R** concludes that $F_i^{\mathbf{R}}$ is an invalid row of T.
3. If **R** has received $t+1$ valid rows, then **R** reconstructs the secret $m^{\mathbf{R}}$ from them and terminates the protocol. Otherwise, **R** proceeds to execute **Phase II**.

Table 1: Matrix T as Computed by $\mathbf{S}$ During **Phase I** of 3-SSMT

$M^{\mathbf{S}}(x)$, Lower order $\frac{n}{3}$ coefficients of $M^{\mathbf{S}}(x)$ are elements of $m^{\mathbf{S}}$			
$M^{\mathbf{S}}(1)$	$M^{\mathbf{S}}(2)$	...	$M^{\mathbf{S}}(n+t)$
$f_1^{\mathbf{S}}(x)$	$f_2^{\mathbf{S}}(x)$	...	$f_{n+t}^{\mathbf{S}}(x)$
$f_1^{\mathbf{S}}(0) = M^{\mathbf{S}}(1)$	$f_2^{\mathbf{S}}(0) = M^{\mathbf{S}}(2)$	...	$f_{n+t}^{\mathbf{S}}(0) = M^{\mathbf{S}}(n+t)$
$f_1^{\mathbf{S}}(1)$	$f_2^{\mathbf{S}}(1)$	...	$f_{n+t}^{\mathbf{S}}(1)$
$f_1^{\mathbf{S}}(2)$	$f_2^{\mathbf{S}}(2)$	...	$f_{n+t}^{\mathbf{S}}(2)$
...	...	...	...
$f_1^{\mathbf{S}}(i)$	$f_2^{\mathbf{S}}(i)$	...	$f_{n+t}^{\mathbf{S}}(i)$
...	...	...	...
$f_1^{\mathbf{S}}(n)$	$f_2^{\mathbf{S}}(n)$	...	$f_{n+t}^{\mathbf{S}}(n)$

Claim 1. *If* $\mathbf{R}$ *concludes that* $F_i^{\mathbf{R}}$ *is a valid row of* T*, then with overwhelming probability* $F_i^{\mathbf{R}} = F_i^{\mathbf{S}}$.

PROOF: The lemma is true without any error if wire f_i is uncorrupted. So let wire f_i be a corrupted wire, who delivers $F_i^{\mathbf{R}} \neq F_i^{\mathbf{S}}$. In this case, if $F_i^{\mathbf{R}}$ is considered as a valid row of T, then it implies that $Support_i \geq t+1$. Since there can be at most t corrupted wires in the top band, this implies that there exists at least one honest wire, say f_j, which correctly and securely delivered the hash key $\alpha_j^{\mathbf{R}} = \alpha_j^{\mathbf{S}}$ and hash value $v_{ij}^{\mathbf{R}} = v_{ij}^{\mathbf{S}} = hash(\alpha_j^{\mathbf{S}}; F_i^{\mathbf{S}}) = hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{S}})$, such that $f_j \in Support_i$. Since $f_j \in Support_i$, it implies that $v_{ij}^{\mathbf{R}} = hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}})$. Since adversary does not know $\alpha_j^{\mathbf{R}}$ and $v_{ij}^{\mathbf{R}}$, he can ensure that $v_{ij}^{\mathbf{R}} = hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{S}})$, as well as $v_{ij}^{\mathbf{R}} = hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}})$, where $F_i^{\mathbf{R}} \neq F_i^{\mathbf{S}}$, with probability at most $\frac{n-1+t}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$, which is negligible in our context. So with very high probability, f_j will not belong to $Support_i$, which is a contradiction. So with overwhelming probability $F_i^{\mathbf{R}} = F_i^{\mathbf{S}}$. □

Claim 2. *During* **Phase I**, *at least* n *coefficients of* $M^{\mathbf{S}}(x)$ *are information theoretically secure.*

PROOF: We consider the worst case, when $\mathcal{A}_t$ controls at most t wires in the top band. Without loss of generality, let these be the first t wires. So $\mathcal{A}_t$ will know the vectors $F_1^{\mathbf{S}}, F_2^{\mathbf{S}}, \ldots, F_t^{\mathbf{S}}$, from which it will come to know t distinct points on the polynomials $f_1^{\mathbf{S}}(x), \ldots, f_{n+t}^{\mathbf{S}}(x)$. But each $f_i^{\mathbf{S}}(x)$ is of degree t and so $\mathcal{A}_t$ will lack by one point to uniquely reconstruct each $f_i^{\mathbf{S}}(x)$. However, $\mathcal{A}_t$ will also know t hash values corresponding to each $F_1^{\mathbf{S}}, \ldots, F_n^{\mathbf{S}}$. Since the vectors $F_1^{\mathbf{S}}, \ldots, F_t^{\mathbf{S}}$ are already known to $\mathcal{A}_t$, the t hash values corresponding to them does not add anything new to $\mathcal{A}_t$'s view. Moreover, the vectors $F_{t+2}^{\mathbf{S}}, \ldots, F_n^{\mathbf{S}}$ can be expressed as a linear combination of vectors $F_1^{\mathbf{S}}, \ldots, F_{t+1}^{\mathbf{S}}$. So the t hash values corresponding to $F_{t+2}^{\mathbf{S}}, \ldots, F_n^{\mathbf{S}}$ can always be expressed as a linear combination of the t hash values corresponding to $F_1^{\mathbf{S}}, \ldots, F_{t+1}^{\mathbf{S}}$, which are known to the adversary. So, out of the t hash values corresponding to each $F_i^{\mathbf{S}}(x), 1 \leq i \leq n$, which are known to $\mathcal{A}_t$, only the t hash values corresponding to $F_{t+1}^{\mathbf{S}}(x)$ add

to $\mathcal{A}_t$'s view. But $F^{\mathbf{S}}_{t+1}$ is of length $n+t$. So from the properties of hashing, $(n+t)-t = n$ coefficients of $F^{\mathbf{S}}_{t+1}$ will be information theoretically secure. This further implies that n coefficients of $M^{\mathbf{S}}(x)$ are information theoretically secure. □

Claim 3. *If* $\mathbf{R}$ *gets* $t+1$ *valid rows of* T *then* $\mathbf{R}$ *can recover* $m^{\mathbf{S}}$.

PROOF: If $\mathbf{R}$ gets $t+1$ valid rows, then from them, $\mathbf{R}$ gets $t+1$ distinct points on each $f^{\mathbf{S}}_i(x)$. Since each $f^{\mathbf{S}}_i(x)$ is of degree t, using the $t+1$ valid rows, $\mathbf{R}$ can reconstruct each $f^{\mathbf{S}}_i(x)$ and hence $f^{\mathbf{S}}_i(0) = M^{\mathbf{S}}(i)$. Now using the $M^{\mathbf{S}}(i)$'s, $\mathbf{R}$ can interpolate $M^{\mathbf{S}}(x)$ and recover $m^{\mathbf{S}}$. □

Lemma 5. *In protocol 3-SSMT if* $\mathbf{R}$ *recovers* $m^{\mathbf{R}}$ *at the end of* **Phase I**, *then with very high probability* $m^{\mathbf{R}} = m^{\mathbf{S}}$. *Moreover,* $\mathcal{A}_t$ *will have no information about* $m^{\mathbf{R}}$.

PROOF: If $\mathbf{R}$ recovers $m^{\mathbf{R}}$ at the end of **Phase I**, then it implies that $\mathbf{R}$ has received $t+1$ valid rows. From Claim 1, all these rows are indeed the rows of T sent by $\mathbf{S}$ with very high probability. So from Claim 3, with very high probability $m^{\mathbf{R}} = m^{\mathbf{S}}$. Moreover, from Claim 2, $\mathcal{A}_t$ will have no information about $m^{\mathbf{R}}$. □

Lemma 6. *If there exists* $t+1$ *honest wires in the top band then* $\mathbf{R}$ *will always be able to recover* $m^{\mathbf{S}}$. *Otherwise, with very high probability,* $\mathbf{R}$ *will detect this. During* **Phase I**, $\mathbf{S}$ *communicates* $\mathcal{O}(n^2\kappa)$ *bits.*

PROOF: If there exists $t+1$ honest wires in the top band then $\mathbf{R}$ will receive $t+1$ valid rows of T over them and hence from Claim 3, $\mathbf{R}$ will correctly recover $m^{\mathbf{R}} = m^{\mathbf{S}}$. On the other hand if there exists less than $t+1$ honest wires in the top band then from the proof of Claim 1, $\mathbf{R}$ will receive less than $t+1$ valid rows with very high probability. So with very high probability, $\mathbf{R}$ will detect that there are at most t honest wires in the top band.

During **Phase I**, $\mathbf{S}$ sends $\Theta(n)$ elements from $\mathbb{F}$ over each wire. So **Phase I** requires a communication complexity of $\mathcal{O}(n^2)$ bits. □

If $\mathbf{R}$ is unable to recover $m^{\mathbf{R}}$ at the end of **Phase I**, then $\mathbf{R}$ concludes that there are at most t honest wires in the top band. This further implies that there exists at least one honest wire in the bottom band. So $\mathbf{R}$ interacts with $\mathbf{S}$ so as to enable $\mathbf{S}$ to re-send $m^{\mathbf{S}}$ using the second method. We now describe second phase of protocol 3-SSMT, which is explained in the next subsection.

5.2.2. Phase II of Protocol 3-SSMT

If the first method to deliver $m^{\mathbf{S}}$ fails at the end of **Phase I**, then in the second method, $\mathbf{S}$ and $\mathbf{R}$ interacts to securely establish an n length vector during **Phase II**. Once this is done, $\mathbf{S}$ can use the elements of the vector as encryption and authentication keys (as in protocol 3-SSMT-Exponential) and using them, $\mathbf{S}$ reliably and securely sends $m^{\mathbf{S}}$ during **Phase III**.

Securely establishing the n length vector is not easy, considering the fact that $\mathbf{S}$ and $\mathbf{R}$ do not know the identity of corrupted wires. Also it is very difficult for $\mathbf{R}$ ($\mathbf{S}$) to reliably send any information to $\mathbf{S}$ ($\mathbf{R}$). In the case of undirected graphs, there exists at least $2t+1$ bi-directional wires between $\mathbf{S}$ and $\mathbf{R}$ (which are necessary and sufficient for the existence of any SRMT/SSMT protocol tolerating $\mathcal{A}_t^{static}$) and so it is very easy to do reliable communication by simply sending the information through all the wires. However here, in the worst case, we may have $t+1$ and t wires in top and bottom band respectively. In protocol 3-SSMT-Exponential, $\mathbf{S}$ and $\mathbf{R}$ could easily establish the vector, as they tried all possible subsets of size $t+1$ and there exists one subset consisting of all honest wires. However, we cannot use the same approach here, as we want to keep the computation and communication complexity polynomial. So we require completely different techniques to reliably and securely establish the keys.

Recall that during **Phase I**, $\mathbf{S}$ has sent an $(n+1)$-tuple over each wire in the top band. These tuples were not used in the first method (during **Phase I**) to send $m^{\mathbf{S}}$. So $\mathbf{R}$ now use these tuples in the second phase. $\mathbf{R}$ does not know which wires in the top band correctly delivered the $(n+1)$-tuples. In order to facilitate $\mathbf{S}$ to find out which tuples were delivered properly, $\mathbf{R}$ hashes each received $(n+1)$-tuple with a random hash key and sends back the hash values and hash keys to $\mathbf{S}$ through the entire bottom band. Since there exists at least one honest wire in the bottom band, it will correctly deliver the hash keys and hash values. In addition to this, $\mathbf{R}$ also sends a random $(n+u)$-tuple over each wire. Furthermore, each $(n+u)$-tuple is hashed by u random hash keys, one corresponding to each wire in the bottom band.

Till now, this part of the computation and communication is some what similar to what is done during protocol 3-SSMT-Exponential. However, there are certain additional steps which are incorporated here. First of all, $\mathbf{S}$ and $\mathbf{R}$ have exchanged tuples over all the wires in top band and bottom band, rather than considering $(t+1)$-sized subsets of wires. In addition to this, $\mathbf{R}$ sends the hash value of each tuple received over the top band and each tuple sent over the bottom band. Finally, the size of the tuples which are exchanged over the top and bottom band are different. They are so to maintain the secrecy property of $m^{\mathbf{S}}$, which will be delivered during third phase. More specifically, each tuple received over the top band is hashed by only one key. So keeping the length of the tuples which are exchanged over top band as $n+1$ maintains the secrecy of its first n elements. On the other hand, each tuple in the bottom band is hashed by u random keys. So keeping the length of the tuples which are exchanged over bottom band as $n+u$ maintains the secrecy of its first n elements.

Now as mentioned above, the information sent by $\mathbf{R}$ may not reach reliably to $\mathbf{S}$. In the worst case, there can be only one honest wire in the bottom band but $\mathbf{S}$ will not know its identity. Now according to the information received from $\mathbf{R}$, $\mathbf{S}$ divides the bottom band into acceptable sets $\mathcal{B}_1, \ldots, \mathcal{B}_k$, where $1 \leq k \leq u$. The division is at the heart of the protocol. Informally, the division is done as follows: $\mathbf{S}$ considers a wire b_i in the bottom band and adds wire b_j from the bottom band in the set $\mathcal{B}_i$ if b_j and b_i are pairwise consistent. Here, by pair

wise consistency we mean that that the hash key and hash value received by **S** over wire b_j is consistent with the $(n+u)$-tuple received by **S** over wire b_i and vice-versa. Notice that if b_j is an honest wire and if b_i and b_j are pairwise consistent, then with very high probability, the $(n+u)$-tuple received by **S** over wire b_i is not modified. This is because the adversary does not know the hash key and hash value sent by **R** over wire b_j, corresponding to the original i^{th} $(n+u)$-tuple.

Now after computing the set $\mathcal{B}_i$, **S** computes the corresponding set $\mathcal{F}_i$. Recall that **R** has sent the hash value of each $(n+1)$-tuple received over the top band, through the entire bottom band. So **S** considers the hash values (corresponding to the top band tuples), received over wire b_i and adds all such f_j's in $\mathcal{F}_i$, such that the $(n+1)$-tuple sent over wire f_j during **Phase I** is consistent with the j^{th} hash value, received over wire b_i. Notice that if f_j is a corrupted wire and if the $(n+1)$-tuple sent over wire f_j is modified during **Phase I**, then with very high probability, f_j will not be added to $\mathcal{F}_i$, provided b_i is an honest wire. This is because if b_i is an honest wire then adversary will not know the j^{th} hash key and hash value, which is sent by **R** over wire b_i.

Finally, **S** considers the set $\mathcal{B}_i$ as acceptable, if there are at least $t+1$ wires in total in $\mathcal{F}_i$ and $\mathcal{B}_i$. It is easy to see that there will be at most u acceptable sets, one corresponding to each wire in the bottom band. Moreover, if b_i is an honest wire in the bottom band, then $\mathcal{B}_i$ will always be an acceptable set, as all honest wires in top band and bottom band will be present in $\mathcal{F}_i$ and $\mathcal{B}_i$ respectively. It is this division of the bottom band, which makes **S** now to work with u path sets, instead of $\binom{2t+1}{t+1}$ path sets. The formal details of **Phase II** of protocol 3-SSMT are given in Fig. 3.

Remark 2. *The three phase efficient SSMT protocol of [14] as well as [34] also divides the bottom band into subsets during second phase, according to some what different criteria. In [14], as well as in [34], the authors claimed that their criteria will create at most u subsets of the bottom band. However, in the subsequent section, we will show that it is not so. In the worst case, there can be $\mathcal{O}(3^u)$ subsets of the bottom band, thus making the communication and computational complexity of their protocol exponential. On the other hand, our criteria for division of bottom band always result in at most u subsets of the bottom band. It is this difference in the criteria of the division of the bottom band, which makes the communication and computational complexity of our protocol 3-SSMT polynomial.*

We now prove the properties of **Phase II** of 3-SSMT.

Claim 4. *If b_i is an honest wire in the bottom band and $b_i \in \mathcal{B}_j$, corresponding to some wire b_j in the bottom band, then with very high probability, the random $(n+u)$-tuple that **S** has received along wire b_j is not modified.*

PROOF: The claim holds without any error if wire b_j is honest. We now show that the claim even holds for a corrupted wire b_j with very high probability. So let b_j be a corrupted wire, such that the $(n+u)$-tuple received by **S** over wire

Figure 3: Phase II of Protocol 3-SSMT

Phase II: R to S: **R** does the following computation and communication:

1. For $i = 1, \ldots, n$, **R** chooses a random non-zero hash key $r_i^{\mathbf{R}} \in \mathbb{F}$, corresponding to wire f_i.
2. **R** computes the set $\beta^{\mathbf{R}} = \{(r_i^{\mathbf{R}}, \gamma_i^{\mathbf{R}}) : i = 1, \ldots, n\}$, where $\gamma_i^{\mathbf{R}} = hash(r_i^{\mathbf{R}}; x_{1,i}^{\mathbf{R}}, \ldots, x_{n+1,i}^{\mathbf{R}})$. Recall that $(x_{1,i}^{\mathbf{R}}, \ldots, x_{n+1,i}^{\mathbf{R}})$ denotes the $(n+1)$-tuple which **R** has received during **Phase I** over wire f_i, for $i = 1, \ldots, n$.
3. For $i = 1, \ldots, u$, **R** selects a random non-zero $(n+u)$-tuple $(y_{1,i}^{\mathbf{R}}, \ldots, y_{n+u,i}^{\mathbf{R}}) \in \mathbb{F}^{n+u}$.
4. Corresponding to the $(n+u)$-tuple $(y_{1,i}^{\mathbf{R}}, \ldots, y_{n+u,i}^{\mathbf{R}})$, **R** selects u random non-zero hash keys $\{key_{i,j}^{\mathbf{R}} : j = 1, \ldots, u\}$ from $\mathbb{F}$.
5. For $i = 1, \ldots, u$, **R** sends the following to **S** through wire b_i:
   (a) $\beta^{\mathbf{R}}$;
   (b) The $(n+u)$-tuple $(y_{1,i}^{\mathbf{R}}, \ldots, y_{n+u,i}^{\mathbf{R}})$;
   (c) The 2-tuple $(key_{j,i}^{\mathbf{R}}, \alpha_{j,i}^{\mathbf{R}})$, where $\alpha_{j,i}^{\mathbf{R}} = hash(key_{j,i}^{\mathbf{R}}; y_{1,j}^{\mathbf{R}}, \ldots, y_{n+u,j}^{\mathbf{R}})$, for $j = 1, \ldots, u$.

Computation by S at the end of Phase II:

1. Let **S** receive the following over wire b_i, for $i = 1, \ldots, u$:
   (a) $\beta_i^{\mathbf{S}} = \{(r_{i,j}^{\mathbf{R}}, \gamma_{i,j}^{\mathbf{R}}) : j = 1, \ldots, n\}$;
   (b) The $(n+u)$-tuple $(y_{1,i}^{\mathbf{S}}, \ldots, y_{n+u,i}^{\mathbf{S}})$;
   (c) The 2-tuple $(key_{j,i}^{\mathbf{S}}, \alpha_{j,i}^{\mathbf{S}})$, for $j = 1, \ldots, u$.
2. For $i = 1, \ldots, u$, corresponding to wire b_i, **S** computes $\mathcal{B}_i$ and $\mathcal{F}_i$ as follows:
   (a) **S** adds wire $b_j \in \{b_1, \ldots, b_u\}$ to $\mathcal{B}_i$ (which is initially $\emptyset$) if b_i, b_j are found to be pair-wise consistent by satisfying both the following conditions:
      i. $\alpha_{i,j}^{\mathbf{S}} = hash(key_{i,j}^{\mathbf{S}}; y_{1,i}^{\mathbf{S}}, \ldots, y_{n+u,i}^{\mathbf{S}})$;
      ii. $\alpha_{j,i}^{\mathbf{S}} = hash(key_{j,i}^{\mathbf{S}}; y_{1,j}^{\mathbf{S}}, \ldots, y_{n+u,j}^{\mathbf{S}})$.
   (b) **S** adds wire f_j to $\mathcal{F}_i$ (which is initially $\emptyset$), if $\gamma_{i,j}^{\mathbf{S}} = hash(r_{i,j}^{\mathbf{S}}; x_{1,j}^{\mathbf{S}}, \ldots, x_{n+1,j}^{\mathbf{S}})$.
   (c) If $|\mathcal{F}_i| + |\mathcal{B}_i| \leq t$ then **S** concludes that $\mathcal{B}_i$ is an unacceptable set, otherwise $\mathcal{B}_i$ is an acceptable set.

b_j is modified; i.e., $(y^{\mathbf{S}}_{1,j}, \ldots, y^{\mathbf{S}}_{n+u,j}) \neq (y^{\mathbf{R}}_{1,j}, \ldots, y^{\mathbf{R}}_{n+u,j})$. Since $b_i \in \mathcal{B}_j$ is an honest wire, it implies that $key^{\mathbf{S}}_{j,i} = key^{\mathbf{R}}_{j,i}$ and $\alpha^{\mathbf{S}}_{j,i} = \alpha^{\mathbf{R}}_{j,i} = hash(key^{\mathbf{R}}_{j,i}; y^{\mathbf{S}}_{1,j}, \ldots, y^{\mathbf{R}}_{n+u,j})$. Moreover, $\mathcal{A}^{static}_t$ will have no information about $(key^{\mathbf{S}}_{j,i}, \alpha^{\mathbf{S}}_{j,i})$. Furthermore, since $b_i \in \mathcal{B}_j$, it implies that $\alpha^{\mathbf{S}}_{j,i} = hash(key^{\mathbf{S}}_{j,i}; y^{\mathbf{S}}_{1,j}, \ldots, y^{\mathbf{S}}_{n+u,j})$. However, from the properties of hashing, without knowing $key^{\mathbf{S}}_{j,i}$, the adversary can ensure that $hash(key^{\mathbf{S}}_{j,i}; y^{\mathbf{S}}_{1,j}, \ldots, y^{\mathbf{S}}_{n+u,j}) = hash(key^{\mathbf{S}}_{j,i}; y^{\mathbf{R}}_{1,j}, \ldots, y^{\mathbf{R}}_{n+u,j})$, even if $(y^{\mathbf{S}}_{1,j}, \ldots, y^{\mathbf{S}}_{n+u,j}) \neq (y^{\mathbf{R}}_{1,j}, \ldots, y^{\mathbf{R}}_{n+u,j})$ with probability at most $\frac{n+u}{|\mathbb{F}|-1} \approx 2^{-\Omega(\kappa)}$, which is negligible in our context. □

Claim 5. *If b_i is an honest wire in the bottom band then $\mathcal{B}_i$ will be always considered as an acceptable set.*

PROOF: Let $f_{i_1}, \ldots, f_{i_T}$ and $b_{i_1}, \ldots, b_{i_B}$ denote the honest wires in the top and bottom band respectively. Now $|\{f_{i_1}, \ldots, f_{i_T}\}| + |\{b_{i_1}, \ldots, b_{i_B}\}| \geq t+1$. Moreover $|\{f_{i_1}, \ldots, f_{i_T}\}| \geq 1$. Furthermore according to the condition given in the claim, $|\{b_{i_1}, \ldots, b_{i_B}\}| \geq 1$. To prove the claim we show that the wires $b_{i_1}, \ldots, b_{i_B}$ will be present in $\mathcal{B}_i$ and the wires $f_{i_1}, \ldots, f_{i_T}$ will be present in the corresponding $\mathcal{F}_i$. So $|\mathcal{F}_i| + |\mathcal{B}_i| = |\{f_{i_1}, \ldots, f_{i_T}\}| + |\{b_{i_1}, \ldots, b_{i_B}\}| \geq t+1$ and thus $\mathcal{B}_i$ will be an acceptable set.

First of all notice that all the wires $b_{i_1}, \ldots, b_{i_B}$ (including b_i) will be present in $\mathcal{B}_i$. This is because any two wires b_j, b_k in the set $\{b_{i_1}, \ldots, b_{i_B}\}$ will be pairwise consistent because the following conditions are satisfied (as both these wires are honest):

1. $\alpha^{\mathbf{S}}_{j,k} = hash(key^{\mathbf{S}}_{j,k}; y^{\mathbf{S}}_{1,j}, \ldots, y^{\mathbf{S}}_{n+u,j})$;
2. $\alpha^{\mathbf{S}}_{k,j} = hash(key^{\mathbf{S}}_{k,j}; y^{\mathbf{S}}_{1,k}, \ldots, y^{\mathbf{S}}_{n+u,k})$.

So the wires $b_{i_1}, \ldots, b_{i_B}$ will be present in some $\mathcal{B}_i$. Since the wires $f_{i_1}, \ldots, f_{i_T}$ are honest, the $(n+1)$-tuple received by $\mathbf{R}$ over these wires are the same as sent by $\mathbf{S}$. That is, $(x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n+1,j}) = (x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$, for every $j \in \{i_1, \ldots, i_T\}$. This implies that $\gamma^{\mathbf{R}}_j = hash(r^{\mathbf{R}}_j; x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n+1,j}) = hash(r^{\mathbf{R}}_j; x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$, for every $j \in \{i_1, \ldots, i_T\}$. Since the wires $b_{i_1}, \ldots, b_{i_B}$ are honest, they will correctly deliver $\beta^{\mathbf{R}}$ and hence $\beta^{\mathbf{S}}_i = \beta^{\mathbf{R}}$, as wire b_i is honest. This implies that $(\gamma^{\mathbf{S}}_{i,j}, r^{\mathbf{S}}_{i,j}) = (\gamma^{\mathbf{R}}_j, r^{\mathbf{R}}_j)$ for every $j \in \{i_1, \ldots, i_T\}$. So when $\mathbf{S}$ executes step 2(b) of the local computation with respect to $\beta^{\mathbf{S}}_i$, all the wires $f_{i_1}, \ldots, f_{i_T}$ will be added in $\mathcal{F}_i$. □

Claim 6. *Let b_i be an honest wire in the bottom band. Then with very high probability, the $(n+1)$-tuple received by $\mathbf{R}$ at the end of* **Phase I** *over the wires in $\mathcal{F}_i$ are not modified.*

PROOF: Let b_i be an honest wire in the bottom band. Then $|\mathcal{F}_i| \geq 1$. This is because from the proof of the previous claim, $|\mathcal{B}_i| + |\mathcal{F}_i| \geq t+1$ and there can be at most t wires in $\mathcal{B}_i$. Now let f_j be an honest wire from the top band, which is present in $\mathcal{F}_i$. Since f_j is honest, it implies that it will correctly deliver the $(n+1)$-tuple to $\mathbf{R}$. On the other hand, let f_j be a corrupted wire

in the top band, such that f_j has modified the $(n+1)$-tuple sent by $\mathbf{S}$ over f_j. That is $(x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n+1,j}) \neq (x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$. We now show that except with probability $2^{-\Omega(\kappa)}$, f_j will not be present in $\mathcal{F}_i$.

Notice that when modifying the $(n+1)$-tuple over wire f_j, the adversary has no idea about the random hash key $r^{\mathbf{R}}_j$, corresponding to wire f_j, which is going to be selected by $\mathbf{R}$ during **Phase II**. Now $\gamma^{\mathbf{R}}_j = hash(r^{\mathbf{R}}_j; x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n+1,j})$. Since wire b_i is honest, it will correctly deliver $\beta^{\mathbf{R}}$ and hence $(r^{\mathbf{R}}_j, \gamma^{\mathbf{R}}_j)$. So $\beta^{\mathbf{S}}_i = \beta^{\mathbf{R}}$ and hence $(r^{\mathbf{S}}_{ij}, \gamma^{\mathbf{S}}_{ij}) = (r^{\mathbf{R}}_j, \gamma^{\mathbf{R}}_j)$. Since $(x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n+1,j}) \neq (x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$ and adversary has no information about $r^{\mathbf{R}}_j$, from the properties of hashing, except with probability $\frac{n}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$, $\gamma^{\mathbf{S}}_{ij} \neq hash(r^{\mathbf{S}}_{ij}; x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$. Thus except with probability $2^{-\Omega(\kappa)}$, f_j will not be present in $\mathcal{F}_i$. □

Claim 7. *Let $\mathcal{B}_i$ be an acceptable set. If b_j is an honest wire in $\mathcal{B}_i$, then the adversary will have no information about the first n values from the $(n+u)$-tuple which is sent by $\mathbf{R}$ over wire b_j. Similarly, if f_j is an honest wire in $\mathcal{F}_i$, then the adversary will have no information about the first n values from the $(n+1)$-tuple which is sent by $\mathbf{S}$ over wire f_j.*

PROOF: Let $\mathcal{B}_i$ be an acceptable set and let b_j be an honest wire in $\mathcal{B}_i$. Since b_j is an honest wire, the adversary will not know the $(n+u)$-tuple which $\mathbf{R}$ will send over wire b_j. Now notice that during **Phase II**, $\mathbf{R}$ hashes the $(n+u)$-tuple which is going to be sent over wire b_j, by u random hash keys and sends one (hash key, hash value) pair through each wire in the bottom band. In the worst case, the entire bottom band, except wire b_j may be under the control of the adversary. So in the worst case, adversary will know $u-1$ distinct hash values, corresponding to the $(n+u)$-tuple sent over wire b_j. So from the properties of hashing, the adversary will have no information about the first n values from the $(n+u)$-tuple which is sent by $\mathbf{R}$ over wire b_j.

Now suppose there exists an honest wire, say f_j, in $\mathcal{F}_i$. So the adversary will not know the $(n+1)$-tuple which $\mathbf{S}$ has sent over wire f_j. Now notice that during **Phase II**, $\mathbf{R}$ hashes the $(n+1)$-tuple received over wire f_j by a random hash key $r^{\mathbf{R}}_j$, adds the pair $(r^{\mathbf{R}}_j, \gamma^{\mathbf{R}}_j)$ to $\beta^{\mathbf{R}}$ and sends $\beta^{\mathbf{R}}$ over all the wires in the bottom band. So even if the entire bottom band is under the control of the adversary, the adversary will know *only one* hash value corresponding to the $(n+1)$-tuple which was sent by $\mathbf{S}$ over wire f_j. So from the properties of hashing, the adversary will have no information about the first n values from the $(n+1)$-tuple which is sent by $\mathbf{S}$ over wire f_j. □

We now summarize the properties of set $\mathcal{B}_i, \mathcal{F}_i$, corresponding to an honest wire b_i in the bottom band.

Lemma 7. *Let b_i be an honest wire in the bottom band. Then $\mathcal{B}_i$ and corresponding $\mathcal{F}_i$ will have the following properties:*

1. *$\mathcal{B}_i$ will be an acceptable set.*

2. *All honest wires in the bottom band will be present in $\mathcal{B}_i$, while all the honest wires in the top band will be present in $\mathcal{F}_i$.*
3. *With very high probability, the $(n+u)$-tuple received by* **S** *at the end of* **Phase II** *over the wires in $\mathcal{B}_i$ are not modified.*
4. *With very high probability, the $(n+1)$-tuple received by* **R** *at the end of* **Phase I** *over the wires in $\mathcal{F}_i$ are not modified.*
5. *The adversary will have no information about the first n values from the $(n+u)$-tuples which are exchanged over the honest wire(s) in $\mathcal{B}_i$. The adversary will also have no information about the first n values from the $(n+1)$-tuples which are exchanged over the honest wires in $\mathcal{F}_i$.*

PROOF: Follows from the proof of Claim 4, Claim 5, Claim 6 and Claim 7. □

We now prove the properties of set $\mathcal{B}_i, \mathcal{F}_i$, corresponding to a wire b_i in the bottom band, such that b_i is under the control of the adversary.

Claim 8. *Let b_i be a corrupted wire in the bottom band, such that there exists an honest wire in $\mathcal{B}_i$. Then the adversary can modify the $(n+u)$-tuple which are exchanged over the corrupted wires in $\mathcal{B}_i$, other than b_i, without letting* **S** *know about it.*

PROOF: Let b_j be an honest wire present in $\mathcal{B}_i$. Then from the proof of Claim 4, the $(n+u)$-tuple is correctly exchanged between **S** and **R** over wire b_i with very high probability. However, it does not imply that the $(n+u)$-tuple are correctly exchanged over other corrupted wires (if any) in $\mathcal{B}_i$. More specifically, let b_k be a corrupted wire in the bottom band, other than b_i, such that $(y^{\mathbf{S}}_{1,k}, \ldots, y^{\mathbf{S}}_{n+u,k}) \neq (y^{\mathbf{R}}_{1,k}, \ldots, y^{\mathbf{R}}_{n+u,k})$. Since b_i is also under the control of the adversary, the adversary will know $(key^{\mathbf{R}}_{k,i}, \alpha^{\mathbf{R}}_{k,i})$. Moreover, the adversary can modify the pair such that $(key^{\mathbf{S}}_{k,i}, \alpha^{\mathbf{S}}_{k,i}) \neq (key^{\mathbf{R}}_{k,i}, \alpha^{\mathbf{R}}_{k,i})$ and $\alpha^{\mathbf{S}}_{k,i} = hash(key^{\mathbf{S}}_{k,i}; y^{\mathbf{S}}_{1,k}, \ldots, y^{\mathbf{S}}_{n+u,k})$. Furthermore, adversary does not modify $(key^{\mathbf{R}}_{i,k}, \alpha^{\mathbf{R}}_{i,k})$ and thus $(key^{\mathbf{S}}_{i,k}, \alpha^{\mathbf{S}}_{i,k}) = (key^{\mathbf{R}}_{i,k}, \alpha^{\mathbf{R}}_{i,k})$, where $\alpha^{\mathbf{S}}_{i,k} = hash(key^{\mathbf{S}}_{i,k}; y^{\mathbf{S}}_{1,i}, \ldots, y^{\mathbf{S}}_{n+u,i})$. If the adversary behaves in this manner, then b_i and b_k will be pairwise consistent, while b_k and b_j will not be pair-wise consistent. But still b_k will be included in $\mathcal{B}_i$ and neither **S** nor **R** will know that the $(n+u)$-tuple exchanged over wire $b_k \in \mathcal{B}_i$ is corrupted. □

Claim 9. *Let b_i be a corrupted wire in the bottom band and let f_j be a corrupted wire in the top band, which is present in $\mathcal{F}_i$. Then the adversary can modify the $(n+1)$-tuple which is exchanged over corrupted wire f_j, without letting* **S** *and* **R** *know about it.*

PROOF: Let f_j be a corrupted wire in the top band, such that $(x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n+1,j}) \neq (x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$. Since b_i is also under the control of the adversary, it implies that $\beta^{\mathbf{R}}$ and hence $(r^{\mathbf{R}}_j, \gamma^{\mathbf{R}}_j)$ is also known to the adversary. Now notice that $\gamma^{\mathbf{R}}_j = hash(r^{\mathbf{R}}_j; x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n+1,j})$. During the transmission of $\beta^{\mathbf{R}}$ over b_i, the adversary can simply change the j^{th} pair in $\beta^{\mathbf{R}}$, such that $(r^{\mathbf{S}}_{i,j}, \gamma^{\mathbf{S}}_{i,j}) \neq (r^{\mathbf{R}}_j, \gamma^{\mathbf{R}}_j)$

and $\gamma^{\mathbf{S}}_{i,j} = hash(r^{\mathbf{S}}_{i,j}; x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$. The adversary can do so because he knows $(x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n+1,j})$ and $r^{\mathbf{R}}_{j}$ and b_i is under his control. So even though the $(n+1)$-tuple is not correctly exchanged over f_j, still f_j can be present in $\mathcal{B}_i$. □

We now summarize the properties of set $\mathcal{B}_i, \mathcal{F}_i$, corresponding to a wire b_i in the bottom band, such that b_i is under the control of the adversary.

Lemma 8. *Let b_i be a wire in the bottom band such that b_i is under the control of the adversary. Moreover, let $\mathcal{B}_i$ be an acceptable set. Then $\mathcal{B}_i$ and corresponding $\mathcal{F}_i$ will have the following properties:*

1. *There will exist at least one honest wire, either from the top band or bottom band, which will be present in $\mathcal{F}_i$ or $\mathcal{B}_i$ respectively.*
2. *If there exists an honest wire in $\mathcal{F}_i$, then the $(n+1)$-tuple is correctly exchanged between $\mathbf{S}$ and $\mathbf{R}$ over that wire. Moreover, adversary will have no information about the first n values of the $(n+1)$-tuple.*
3. *If there exists an honest wire in $\mathcal{B}_i$, then the $(n+u)$-tuple is correctly exchanged between $\mathbf{S}$ and $\mathbf{R}$ over that wire. Moreover, adversary will have no information about the first n values of the $(n+u)$-tuple.*
4. *If there exists an honest wire in $\mathcal{B}_i$, then with very high probability the $(n+u)$-tuple is correctly exchanged between $\mathbf{S}$ and $\mathbf{R}$ over wire b_i. However, the adversary can modify the $(n+u)$-tuple which are exchanged over the corrupted wires in $\mathcal{B}_i$, other than b_i, without letting $\mathbf{S}$ know about it.*
5. *If there is no honest wire in $\mathcal{B}_i$, then the adversary can always modify the $(n+u)$-tuple which are exchanged over the corrupted wires in $\mathcal{B}_i$, without letting $\mathbf{S}$ know about it.*
6. *Irrespective of the number of honest wires in the top band, the adversary can always modify the $(n+1)$-tuple which is exchanged over a corrupted wire (if any) in $\mathcal{F}_i$, without letting $\mathbf{S}$ and $\mathbf{R}$ know about it.*

PROOF: Since $\mathcal{B}_i$ is an acceptable set, it implies that $|\mathcal{B}_i| + |\mathcal{F}_i| \geq t+1$. In the worst case there can t corrupted wires including top and bottom band. This implies that there exists at least one honest wire, which is present either in $\mathcal{F}_i$ or $\mathcal{B}_i$. This proves the first property.

Let f_j be an honest wire present in $\mathcal{F}_i$. It implies that the $(n+1)$-tuple is correctly exchanged between $\mathbf{S}$ and $\mathbf{R}$ over wire f_j. Moreover, from the proof of Claim 7, adversary will have no information about the first n values of the $(n+1)$-tuple. This proves the second property.

Let b_j be an honest wire present in $\mathcal{B}_i$. It implies that the $(n+u)$-tuple is correctly exchanged between $\mathbf{S}$ and $\mathbf{R}$ over wire b_j. Moreover, from the proof of Claim 7, adversary will have no information about the first n values of the $(n+u)$-tuple. This proves the third property.

The fourth property follows from the proof of Claim 8.

We now prove the fifth property. The corrupted wires in the bottom band can behave in such a way that even though there does not exist any honest wire in $\mathcal{B}_i$, still $\mathcal{B}_i$ becomes an acceptable set. We consider a possible setting and

adversarial behavior in which it is possible. The setting can be easily generalized. More specifically, suppose $u = t$ and there exists $t-1$ corrupted wires in the bottom band, who modify the $(n+u)$-tuple exchanged over them. Moreover, as explained in the proof of Claim 8, the adversary can control these $t-1$ wires in such a way that from $\mathbf{S}$'s point of view, all the $t-1$ corrupted wires are pairwise consistent. Furthermore, adversary can ensure that none of these $t-1$ corrupted wires are pair wise consistent with the single honest wire which is present in the bottom band. That is, if b_j is an honest wire in the bottom band, then the adversary can simply modify $(key^{\mathbf{R}}_{j,k}, \alpha^{\mathbf{R}}_{j,k})$, over all $t-1$ corrupted b_k's, thus making b_j not pair-wise consistent with any of the $t-1$ corrupted wires. Now as a result of such adversarial behavior, only corrupted wires $(t-1)$ will be present in $\mathcal{B}_i$. In order that $\mathcal{B}_i$ becomes acceptable, there should be at least two wires in $\mathcal{F}_i$. Notice that in the scenario we are considering, there are t honest wires in the top band, who will correctly deliver the $(n+1)$-tuples to $\mathbf{R}$. So if b_i correctly delivers $\beta^{\mathbf{R}}$ without doing any modification, then $\beta^{\mathbf{S}}_i = \beta^{\mathbf{R}}$ and hence $\mathbf{S}$ will include all the t honest wires in the top band in $\mathcal{F}_i$ and hence $\mathcal{B}_i$ will become acceptable. This proves the fifth property.

The last property follows from the proof of Claim 9. □

Remark 3 (Difference Between Corrupted and Honest Acceptable Set). *Comparing Lemma 7 and Lemma 8, we find that in case of honest b_i, the $(n+1)$-tuples and $(n+u)$-tuples are exchanged correctly between $\mathbf{S}$ and $\mathbf{R}$ over all the wires in $\mathcal{F}_i$ and $\mathcal{B}_i$ respectively with very high probability. Moreover, there will be at least $t+1$ (honest) wires distributed in $\mathcal{F}_i$ and $\mathcal{B}_i$, such that adversary will have no information about the first n values of the tuples that are exchanged over those wires. On the other hand, in the case of corrupted b_i, the $(n+1)$-tuples and $(n+u)$-tuples are exchanged correctly between $\mathbf{S}$ and $\mathbf{R}$ only over the honest wires in $\mathcal{F}_i$ and $\mathcal{B}_i$ respectively. Moreover, there will be at least one (honest) wire either in $\mathcal{F}_i$ or $\mathcal{B}_i$, such that the adversary will have no information about the first n values of the tuple that is exchanged over that wire.*

Lemma 9. *In protocol 3-SSMT, there can be at most u acceptable sets.*

PROOF: The proof follows from the fact corresponding to each wire b_i in the bottom band, there is only one $\mathcal{B}_i$ and $\mathcal{F}_i$. □

Lemma 10. *In protocol 3-SSMT, $\mathbf{R}$ communicates $\mathcal{O}(n^2\kappa)$ bits during* **Phase II**.

PROOF: During **Phase II**, $\mathbf{R}$ sends $\mathcal{O}(n)$ field elements over each wire. This requires a communication complexity of $\mathcal{O}(nu\kappa) = \mathcal{O}(n^2\kappa)$ bits, as $u = \mathcal{O}(n)$. □

Finally before proceeding further to the description of third phase of the protocol, we note that $\mathbf{R}$ executes **Phase II** only if he could not recover $m^{\mathbf{S}}$ at the end of **Phase I**. However, even if $\mathbf{R}$ recovers $m^{\mathbf{S}}$ at the end of **Phase I**, he has no way to signal this to $\mathbf{S}$. This is because there are at most t wires in the bottom band and in the worst case, entire bottom band may be under the control of the

adversary. So even if $\mathbf{R}$ does not execute **Phase II** and does no communication to $\mathbf{S}$, in the worst case the adversary may simply control some wire(s) in the bottom band and passes some information to $\mathbf{S}$, which may lead $\mathbf{S}$ to construct some valid acceptable set(s), corresponding to those wire(s)!! Fortunately the properties of hashing ensures that this happens with very negligible probability, as shown in the next lemma.

Lemma 11. *Suppose in protocol 3-SSMT,* $\mathbf{R}$ *recovers* $m^{\mathbf{S}}$ *at the end of* **Phase I** *and does no communication over the bottom band. Then the probability that the adversary corrupts bottom band and sends some arbitrary information to* $\mathbf{S}$ *which leads to the construction of acceptable set is at most* $2^{-\Omega(\kappa)}$.

PROOF: Consider the following settings: there are $t+2$ wires in the top band and $t-1$ wires in the bottom band, such that the entire bottom band and one wire from the top band are under the control of the adversary. Suppose $\mathbf{R}$ recovers $m^{\mathbf{S}}$ at the end of **Phase I** itself and so does not execute **Phase II**. However, the adversary may do the following: the adversary selects $t-1$ arbitrary $(n+u)$-tuples and $(t-1)^2$ random hash keys and put these values over the wires in the bottom band in such a way, as if $\mathbf{R}$ has executed **Phase II** using the $(n+u)$-tuples and $(t-1)^2$ random hash keys. So it is easy to see that all the wires in the bottom band will be pairwise consistent and thus $|\mathcal{B}_i| = t-1$, for $i = 1, \ldots, t-1$. However, in order that any of these $\mathcal{B}_i$ becomes an acceptable set, the adversary has to ensure that the corresponding $|\mathcal{F}_i| \geq 2$.

Now notice that the adversary will know the $(n+1)$-tuple which is exchanged over the single wire in the top band which is under the control of the adversary. So the adversary can *always* produce the hash value of this tuple, corresponding to any hash key. However, the adversary will have no information about the $(n+1)$-tuples which are exchanged over the $t+1$ honest wires in the top band. The probability that the adversary will be able to produce the hash value of the $(n+1)$-tuple, corresponding to any of these $t+1$ honest wires, for a given hash key is same as the probability of correctly guessing the corresponding $(n+1)$-tuple, which is $\frac{n+1}{|\mathbb{F}|^{n+1}} \approx 2^{-\Omega(\kappa)}$.

Now the adversary may do the following: he selects $t+2$ hash keys, one corresponding to each wire in the top band. The adversary also guesses the $(n+1)$-tuple which $\mathbf{S}$ would have sent over each honest wire in the top band and computes the hash value of those tuples, as if the tuples are received by $\mathbf{R}$. The adversary also computes the hash value of the $(n+1)$-tuple, which $\mathbf{S}$ has sent over the wire under its control in the top band. Thus the adversary computes $\beta^{\mathbf{R}}$, as if $\beta^{\mathbf{R}}$ is computed by $\mathbf{R}$. The adversary is sure that at least one (hash-key, hash-value) pair in the computed $\beta^{\mathbf{R}}$, namely the one corresponding to the corrupted wire in the top band is correct. The remaining (hash-key, hash-value) pair in the computed $\beta^{\mathbf{R}}$ (corresponding to the honest wires in the top band) may be correct, depending upon the guess of the adversary. The adversary then sends the computed $\beta^{\mathbf{R}}$ to $\mathbf{S}$ over the entire bottom band. On receiving $\beta_i^{\mathbf{S}} = \beta^{\mathbf{R}}$, $\mathbf{S}$ will add the corrupted wire in the top band to the set $\mathcal{F}_i$. Moreover, if the adversary has successfully guessed the $(n+1)$-tuple which was

sent over some honest wire f_j, then the j^{th} (hash-key, hash-value) pair in $\beta_i^{\mathbf{S}}$ would be correct and hence f_j will be added in $\mathcal{F}_i$, thus making $|\mathcal{F}_i| = 2$ and hence $\mathcal{B}_i$ as an acceptable set. However, this happens with probability $2^{-\Omega(\kappa)}$. □

We now proceed towards the discussion of third phase, which is given in the next subsection.

5.3. Phase III of Protocol 3-SSMT

Notice that at the end of second phase, **S** could have u acceptable sets. **S** will not know which of these acceptable sets contains only honest wires. From **S**'s view point, all acceptable set may look valid. So **S** assumes that all the acceptable sets are valid and tries to compute separate encryption key and authentication key from each acceptable set. Specifically, **S** considers the wires in each acceptable $\mathcal{B}_i$ and corresponding $\mathcal{F}_i$ as a valid path set and using them, **S** does the same computation and communication as in **Phase III** of protocol 3-SSMT-Exponential. However, instead of dealing with $\binom{2t+1}{t+1}$ path sets, **S** has to only consider u path sets. In the same way, **R** recovers the message at the end of third phase by performing similar computation, as it does at the end of **Phase III** of 3-SSMT-Exponential.

The secrecy of the protocol follows from the fact that corresponding to every acceptable $\mathcal{B}_i$, there exists at least one honest wire, either in $\mathcal{B}_i$ or corresponding $\mathcal{F}_i$, such that the adversary will have no information about the first n values of the tuple which is exchanged correctly between **S** and **R** over the honest wire. So adversary will have no information about the n-tuple (whose elements are considered as the authentication and encryption keys), computed by **S** from the tuples, which are exchanged over the wires in $\mathcal{B}_i$ and $\mathcal{F}_i$. Moreover, if the tuples are not correctly exchanged between **S** and **R** over the wires in $\mathcal{B}_i$ and $\mathcal{F}_i$, then **S** and **R** will end up with different version of authentication and encryption keys. So except with negligible error probability, the verification at **R**'s end will fail. This ensures reliability. The **Phase III** of protocol 3-SSMT is formally presented in Fig. 4.

We now prove the properties of **Phase III** of protocol 3-SSMT.

Lemma 12. *In protocol 3-SSMT, $m^{\mathbf{S}}$ will be information theoretically secure at the end of* **Phase III**.

Proof: In protocol 3-SSMT, during **Phase III**, **S** encrypts and authenticates $m^{\mathbf{S}}$ by using the tuples, which are exchanged between **S** and **R** over the wires in $\mathcal{B}_l$ and $\mathcal{F}_l$, if $\mathcal{B}_l$ is an acceptable set. Now as stated in Remark 3, irrespective of whether wire b_l is honest or corrupted, there exists at least one honest wire, either in $\mathcal{B}_l$ or $\mathcal{F}_l$, such that the adversary will have no information about the first n elements of the tuple which is exchanged between **S** and **R** over that honest wire. Since the keys $\mathcal{C}_{1,l}^{\mathbf{S}}, \ldots, \mathcal{C}_{n,l}^{\mathbf{S}}$ are computed by adding the first n elements of the all the tuples which are exchanged between **S** and **R** over the wires in $\mathcal{B}_l$ or $\mathcal{F}_l$, it implies that $\mathcal{C}_{1,l}^{\mathbf{S}}, \ldots, \mathcal{C}_{n,l}^{\mathbf{S}}$ will be information theoretically secure. Since $\mathcal{C}_{1,l}^{\mathbf{S}}, \ldots, \mathcal{C}_{n,l}^{\mathbf{S}}$ are used as encryption and authentication keys, by

Figure 4: Phase III of Protocol 3-SSMT

Phase III: S to R: For each acceptable set $\mathcal{B}_l$ and corresponding set $\mathcal{F}_l$, **S** does the following computation and communication:

1. **S** considers the first n elements from the $(n+1)$-tuples which it had sent over the wires in $\mathcal{F}_l$ during **Phase I** and the first n elements from the $(n+u)$- tuples which **S** has received over the wires in $\mathcal{B}_l$ during **Phase II**. By using them, **S** computes his version of n keys $\mathcal{C}^{\mathbf{S}}_{1,l} = \sum_{f_j \in \mathcal{F}_l} x^{\mathbf{S}}_{1,j} + \sum_{b_j \in \mathcal{B}_l} y^{\mathbf{S}}_{1,j}$, $\mathcal{C}^{\mathbf{S}}_{2,l} = \sum_{f_j \in \mathcal{F}_l} x^{\mathbf{S}}_{2,j} + \sum_{b_j \in \mathcal{B}_l} y^{\mathbf{S}}_{2,j}$, ..., $\mathcal{C}^{\mathbf{S}}_{n,l} = \sum_{f_j \in \mathcal{F}_l} x^{\mathbf{S}}_{n,j} + \sum_{b_j \in \mathcal{B}_l} y^{\mathbf{S}}_{n,j}$.
2. For each element of $m^{\mathbf{S}}$ (recall that $|m^{\mathbf{S}}| = \frac{n}{3}$), **S** takes three elements from the keys computed in the previous step and computes the set $\mathcal{S}^{\mathbf{S}}_l = \{(c^{\mathbf{S}}_{i,l}, d^{\mathbf{S}}_{i,l}) : i = 1, \ldots, \frac{n}{3}\}$ where $(c^{\mathbf{S}}_{i,l}, d^{\mathbf{S}}_{i,l}) = USauth(m^{\mathbf{S}}_i; \mathcal{C}^{\mathbf{S}}_{3i-2,l}, \mathcal{C}^{\mathbf{S}}_{3i-1,l}, \mathcal{C}^{\mathbf{S}}_{3i,l})$, for $i = 1, \ldots, \frac{n}{3}$.
3. **S** sends the set $\mathcal{F}_l, \mathcal{B}_l$ and $\mathcal{S}^{\mathbf{S}}_l$ to **R** over all the wires in the set $\mathcal{F}_l$ and terminates the protocol.

Message Recovery by R:

1. Let **R** receive the sets $\mathcal{F}^{\mathbf{R}}_{j,l}, \mathcal{B}^{\mathbf{R}}_{j,l}$ and $\mathcal{S}^{\mathbf{R}}_{j,l}$ along wire f_j, for $j = 1, \ldots, n$. In the worst case, $l = 1, \ldots, u$. **R** then executes the following steps.
2. If for some $j \in \{1, 2, \ldots, n\}$ and some $l \in \{1, 2, \ldots, u\}$, $|\mathcal{F}^{\mathbf{R}}_{j,l}| + |\mathcal{B}^{\mathbf{R}}_{j,l}| \leq t$, then **R** concludes that wire f_j is corrupted and neglects all the values received along f_j.
3. If f_j is not neglected, then for each $\mathcal{F}^{\mathbf{R}}_{j,l}, \mathcal{B}^{\mathbf{R}}_{j,l}$ and $\mathcal{S}^{\mathbf{R}}_{j,l}$ received along wire f_j, **R** does the following:
   (a) Let $\mathcal{S}^{\mathbf{R}}_{j,l} = \{(c^{\mathbf{R}}_{j,i,l}, d^{\mathbf{R}}_{j,i,l}) : i = 1, \ldots, \frac{n}{3}\}$.
   (b) By using the index of the wires in $\mathcal{F}^{\mathbf{R}}_{j,l}$ and $\mathcal{B}^{\mathbf{R}}_{j,l}$, **R** computes his version of n keys $\mathcal{C}^{\mathbf{R}}_{j,1,l}, \ldots, \mathcal{C}^{\mathbf{R}}_{j,n,l}$.
   (c) For $i = 1, \ldots, \frac{n}{3}$, **R** applies the verification process of $USauth$ on $c^{\mathbf{R}}_{j,i,l}, d^{\mathbf{R}}_{j,i,l}, \mathcal{C}^{\mathbf{R}}_{j,3i-2,l}, \mathcal{C}^{\mathbf{R}}_{j,3i-1,l}$ and $\mathcal{C}^{\mathbf{R}}_{j,3i,l}$.
   (d) If the verification is successful for all $i = 1, \ldots, \frac{n}{3}$, then **R** recovers $m^{\mathbf{R}}_{j,i,l}$ from $c^{\mathbf{R}}_{j,i,l}$, for $i = 1, \ldots, \frac{n}{3}$.
   (e) Finally, **R** concatenates $m^{\mathbf{R}}_{j,1,l}, \ldots, m^{\mathbf{R}}_{j,\frac{n}{3},l}$ to reconstruct the secret $m^{\mathbf{R}}$ and terminates the protocol.

the properties of $USauth$, it follows that $m^{\mathbf{S}}$ will be information theoretically secure at the end of **Phase III**. □

Lemma 13. *In protocol 3-SSMT, $m^{\mathbf{R}} = m^{\mathbf{S}}$ with very high probability.*

PROOF: Let $\mathcal{F}_{j,l}^{\mathbf{R}}, \mathcal{B}_{j,l}^{\mathbf{R}}$ and $\mathcal{S}_{j,l}^{\mathbf{R}}$ denote the sets, which passes the verification test in step 3 of message recovery. This implies that $\mathbf{R}$ has recovered $m^{\mathbf{R}}$ from $\mathcal{S}_{j,l}^{\mathbf{R}}$ after computing his keys from the tuples which are exchanged between $\mathbf{S}$ and $\mathbf{R}$ over the wires in $\mathcal{F}_{j,l}^{\mathbf{R}}$ and $\mathcal{B}_{j,l}^{\mathbf{R}}$. Now notice that there exists at least one honest wire, which is present in either $\mathcal{F}_{j,l}^{\mathbf{R}}$ or $\mathcal{B}_{j,l}^{\mathbf{R}}$, such that adversary will have no information about the first n elements of the tuple exchanged over that wire. So the keys computed by $\mathbf{R}$ from $\mathcal{F}_{j,l}^{\mathbf{R}}$ and $\mathcal{B}_{j,l}^{\mathbf{R}}$ will be information theoretically secure. Moreover, as explained in the previous lemma, the keys which are used by $\mathbf{S}$ for encrypting and authenticating $m^{\mathbf{S}}$ will also be information theoretically secure. Now irrespective of whether the tuples are correctly exchanged between $\mathbf{S}$ and $\mathbf{R}$ over all the wires in $\mathcal{F}_{j,l}^{\mathbf{R}}$ and $\mathcal{B}_{j,l}^{\mathbf{R}}$, the adversary will have no information about the keys used by $\mathbf{S}$ and the keys used by $\mathbf{R}$. So from the properties of $USauth$, if at all $\mathbf{R}$ outputs $m^{\mathbf{R}}$ from $\mathcal{S}_{j,l}^{\mathbf{R}}$, then except with probability $2^{-\Omega(\kappa)}$, $m^{\mathbf{R}} = m^{\mathbf{S}}$. □

Lemma 14. *In protocol 3-SSMT, there always exist a wire f_j in the top band, such that $\mathcal{F}_{j,l}^{\mathbf{R}}, \mathcal{B}_{j,l}^{\mathbf{R}}$ and $\mathcal{S}_{j,l}^{\mathbf{R}}$ received by $\mathbf{R}$ over wire f_j will satisfy the verification process.*

PROOF: The proof simply follows from the fact that there exists at least $t+1$ honest wires including top and bottom band and these $t+1$ honest wires will form some acceptable set $\mathcal{B}_l$ and corresponding $\mathcal{F}_l$. The rest now follows from the protocol code for **Phase III**. □

Lemma 15. *During* **Phase III**, $\mathbf{S}$ *communicates $\mathcal{O}(n^3\kappa)$ bits.*

PROOF: During **Phase III**, corresponding to an acceptable set $\mathcal{B}_l$, $\mathbf{S}$ sends the identity of $\mathcal{B}_l, \mathcal{F}_l$ and the encryption, authentication of $m^{\mathbf{S}}$ along all the wires in $\mathcal{F}_l$. This requires a communication complexity of $\mathcal{O}(n^2)$ field elements and hence $\mathcal{O}(n^2\kappa)$ bits. Now there can be $u = \mathcal{O}(n)$ acceptable sets. So the worst case communication complexity of **Phase III** is $\mathcal{O}(n^3\kappa)$ bits. □

We now summarize the properties of protocol 3-SSMT by the following theorem.

Theorem 2. *Protocol 3-SSMT is valid SSMT protocol, which sends a message containing $\frac{n}{3}\kappa$ bits by communicating $\mathcal{O}(n^3\kappa)$ bits and takes at most three phases.*

Finally, before ending our discussion on three phase SSMT, we discuss about the three phase SSMT protocol of [14] and [34] and show that the computational and communication complexity of both the protocols are exponential, as opposed to polynomial, as claimed in [14] and [34].

5.4. Inefficiency of the Three Phase SSMT of [14, 34, 55]

In [34], the authors presented a three phase SSMT protocol called $\Pi^{Existing}_{modified}$ (see Page 314 of [34]). The protocol securely sends a message containing $\frac{n}{3}$ field elements. The idea of the protocol is similar to the three phase SSMT protocol of [14, 55] [6] except that the SSMT protocol of [14, 55] sends a single message. The authors in [34] called the three phase SSMT protocol of [14, 55] as $\Pi^{Existing}$. The authors in [34], as well as in [14, 55] claimed that their three phase SSMT protocol requires polynomial computational and communication complexity [7]. However, we now show that the computational and communication complexity of the SSMT protocols of [14, 34, 55] are exponential.

We specifically consider protocol $\Pi^{Existing}_{modified}$ and show an adversarial behavior, which may result **S** to communicate exponential number of bits during third phase. The behavior also causes **S** and **R** to perform exponential computation during **Phase III** and at the end of **Phase III** respectively. A similar behavior can cause the same result for the SSMT protocol $\Pi^{Existing}$.

The three phase SSMT protocol $\Pi^{Existing}_{modified}$ of [34] is same as protocol 3-SSMT presented in the previous section, except for the computation which is done by **S** at the end of **Phase II**. Specifically, in protocol $\Pi^{Existing}_{modified}$, **S** divides the bottom band at the end of **Phase II**, using some what different criteria, as shown in Fig. 5.

In [34], the authors claimed that there can be at most u acceptable sets, one corresponding to each wire in the bottom band. However, we now show that this is not true. In fact, we show that in the worst case there can be $\mathcal{O}(3^t)$ acceptable sets. Before doing so, we first present few concepts from graph theory.

Definition 9 (Maximal Clique and Maximal Independent Set [1, 2]). *A maximal clique in a graph is a clique that cannot be extended by adding one more vertex to the clique. Complimentarily a maximal independent set is an independent set which cannot be extended by adding one more vertex to the independent set. If S is a maximal independent set in some graph, then it is a maximal clique in the complementary graph.*

Definition 10 (Tuŕan Graph[3]). *The Tuŕan Graph $T(n, r)$ is a graph formed by partitioning a set of n vertices into r subsets, with sizes as equal as possible, and connecting two vertices by an edge whenever they belong to different subsets. That is, it is a complete r-partite graph*

$$K_{\lceil n/r \rceil, \lceil n/r \rceil, \ldots, \lfloor n/r \rfloor, \lfloor n/r \rfloor}$$

The following result from [31] states the upper bound on the number of maximal cliques that can be possible in any graph.

[6] The three phase SSMT protocol presented in [14] and [55] are same.

[7] [34] claimed that their three phase SSMT protocol requires a communication complexity of $\mathcal{O}(n^3\kappa)$ bits, where as [14, 55] claimed that their SSMT protocol is efficient, requiring polynomial computational and communication complexity.

Figure 5: Computation by $\mathbf{S}$ at the End of Phase II of Protocol $\Pi^{Existing}_{modified}$ [34]

S does the Following Computation at the End of Phase II:

1. Let $\mathbf{S}$ receive the following over wire b_i, for $i = 1, \ldots, u$:
   (a) $\beta_i^{\mathbf{S}} = \{(r_{i,j}^{\mathbf{R}}, \gamma_{i,j}^{\mathbf{R}}) : j = 1, \ldots, n\}$;
   (b) The $(n+u)$-tuple $(y_{1,i}^{\mathbf{S}}, \ldots, y_{n+u,i}^{\mathbf{S}})$;
   (c) The 2-tuple $(key_{j,i}^{\mathbf{S}}, \alpha_{j,i}^{\mathbf{S}})$, for $j = 1, \ldots, u$.
2. $\mathbf{S}$ divides the bottom band $\{b_1, \ldots, b_u\}$ into subsets $\mathcal{B}_1, \ldots, \mathcal{B}_k$, such that for $l = 1, \ldots, k$, every two wires $b_i, b_j \in \mathcal{B}_l$ are pair-wise consistent by satisfying the following conditions:
   (a) $\beta_i^{\mathbf{S}} = \beta_j^{\mathbf{S}}$;
   (b) $\alpha_{i,j}^{\mathbf{S}} = hash(key_{i,j}^{\mathbf{S}}; y_{1,i}^{\mathbf{S}}, \ldots, y_{n+u,i}^{\mathbf{S}})$;
   (c) $\alpha_{j,i}^{\mathbf{S}} = hash(key_{j,i}^{\mathbf{S}}; y_{1,j}^{\mathbf{S}}, \ldots, y_{n+u,j}^{\mathbf{S}})$.
3. For $l = 1, \ldots, k$, $\mathbf{S}$ computes the set $\mathcal{F}_l$, corresponding to $\mathcal{B}_l$ as follows:
   (a) Let $b_i \in \mathcal{B}_l$.
   (b) $\mathbf{S}$ adds wire f_j in $\mathcal{F}_l$ if $\gamma_{i,j}^{\mathbf{S}} = hash(r_{i,j}^{\mathbf{S}}; x_{1,j}^{\mathbf{S}}, \ldots, x_{n+1,j}^{\mathbf{S}})$.
4. For $l = 1, \ldots, k$, if $|\mathcal{F}_l| + \mathcal{B}_l \geq t + 1$ then $\mathbf{S}$ considers $\mathcal{B}_l$ as an acceptable set, otherwise $\mathbf{S}$ considers $\mathcal{B}_l$ as unacceptable.

Theorem 3 ([31, 1, 2]). *Any graph with n vertices can have at most* $3^{\frac{n}{3}}$ *maximal cliques.*

The Tuŕan graph $T(n, \lceil n/3 \rceil)$, also called as **Moon-Moser** graph satisfies the bound given in the above theorem, as shown in the following example:

Example 1 (Largest Number of Maximal Cliques Possible in a Graph). *Let G be a graph with n vertices, which is a disjoint union of* $n/3$ *triangle graphs. Any maximal independent set in this graph is formed by choosing one vertex from each triangle. It is easy to see that there will be* $3^{n/3}$ *maximal independent sets in G. Moreover, these maximal independent sets will be maximal clique in the complementary graph* $\overline{G}$*. Thus* $\overline{G}$ *will have exactly* $3^{n/3}$ *maximal cliques. The graph* $\overline{G}$ *is nothing, but the Tuŕan graph* $T(n, \lceil n/3 \rceil)$*, which is also called as* **Moon-Moser** *graph.*

Now we return back to the computation done by **S** at the end of **Phase II** in protocol $\Pi^{Existing}_{modified}$, as given in Fig. 5. We define the following graph:

Definition 11 (Consistency Graph). *Let* $G = (V, E)$ *be an undirected graph where* $V = \{b_1, \ldots, b_u\}$ *and* $(b_i, b_j) \in E$ *iff* b_i, b_j *are pairwise consistent and satisfies the following conditions:*

1. $\beta_i^{\mathbf{S}} = \beta_j^{\mathbf{S}}$;
2. $\alpha_{i,j}^{\mathbf{S}} = hash(key_{i,j}^{\mathbf{S}}; y_{1,i}^{\mathbf{S}}, \ldots, y_{n+u,i}^{\mathbf{S}})$;
3. $\alpha_{j,i}^{\mathbf{S}} = hash(key_{j,i}^{\mathbf{S}}; y_{1,j}^{\mathbf{S}}, \ldots, y_{n+u,j}^{\mathbf{S}})$.

Then the graph G is called the consistency graph.

We next claim that each $\mathcal{B}_l$ computed by **S** in $\Pi^{Existing}_{modified}$ is a maximal clique in the consistency graph G.

Claim 10. *Each* $\mathcal{B}_l$ *computed by* **S** *in Fig. 5 is a maximal clique in consistency graph G.*

Proof: Follows from the definition of maximal clique, consistency graph and the steps executed to compute $\mathcal{B}_l$. □

Claim 11. *Let* b_i *be a wire in the bottom band which is under the control of the adversary and let* b_j *be another wire in the bottom band (other than* b_i*). Then irrespective of whether* b_j *is under the control of the adversary or not, the adversary can control the behavior of* b_i *during* **Phase II** *of* $\Pi^{Existing}_{modified}$ *and decide whether* b_i *is consistent with* b_j *or not.*

Proof: First of all, in order that b_i, b_j are pair-wise consistent, the following must hold:

1. $\beta_i^{\mathbf{S}} = \beta_j^{\mathbf{S}}$;
2. $\alpha_{i,j}^{\mathbf{S}} = hash(key_{i,j}^{\mathbf{S}}; y_{1,i}^{\mathbf{S}}, \ldots, y_{n+u,i}^{\mathbf{S}})$;

3. $\alpha^{\mathbf{S}}_{j,i} = hash(key^{\mathbf{S}}_{j,i}; y^{\mathbf{S}}_{1,j}, \ldots, y^{\mathbf{S}}_{n+u,j})$.

Since b_i is under the control of the adversary, the following information is completely under his control:

1. $\beta^{\mathbf{S}}_i$;
2. The $(n+u)$-tuple $(y^{\mathbf{S}}_{1,i}, \ldots, y^{\mathbf{S}}_{n+u,i})$;
3. The pairs $(key^{\mathbf{S}}_{j,i}; \alpha^{\mathbf{S}}_{j,i})$, for all $j = 1, \ldots, u$.

Now let b_j be a specific wire in the bottom band (other than b_i). If b_j is also under the control of the adversary, then the adversary can always control b_i, b_j and decide whether b_i, b_j are consistent or inconsistent. On the other hand, even if b_j is honest, the adversary can control b_i and decide whether b_i are b_j are consistent or inconsistent. Specifically, if adversary wants that b_i and b_j are consistent, then the adversary does not modify the information passed over wire b_i. That is, $\beta^{\mathbf{S}}_i = \beta^{\mathbf{R}}_i$, $(y^{\mathbf{S}}_{1,i}, \ldots, y^{\mathbf{S}}_{n+u,i}) = (y^{\mathbf{R}}_{1,i}, \ldots, y^{\mathbf{R}}_{n+u,i})$ and $(key^{\mathbf{S}}_{j,i}; \alpha^{\mathbf{S}}_{j,i}) = (key^{\mathbf{R}}_{j,i}; \alpha^{\mathbf{R}}_{j,i})$. Since b_j is anyway honest, this implies that b_i, b_j will be pairwise consistent.

On the other hand, suppose adversary wants b_i, b_j to be inconsistent. The adversary can do so by arbitrarily changing $(key^{\mathbf{S}}_{j,i}; \alpha^{\mathbf{S}}_{j,i})$, such that $(key^{\mathbf{S}}_{j,i}; \alpha^{\mathbf{S}}_{j,i}) \neq (key^{\mathbf{R}}_{j,i}; \alpha^{\mathbf{R}}_{j,i})$. In this case, b_i, b_j will not not be consistent. □

Lemma 16. *The adversary can behave during* **Phase II** *of protocol* $\Pi^{Existing}_{modified}$ *in such a way that it results in* $\mathcal{O}(3^t)$ *acceptable sets.*

Proof: To prove the lemma, we consider the following specific network settings and adversarial behavior. However, the settings and the behavior can be easily generalized. Suppose $n = t+1$ and $u = t$. Moreover, without loss of generality, let $f_1, \ldots, f_t$ and b_t be the honest wires in the top band and bottom band respectively. Furthermore, let f_{t+1} and $b_1, \ldots, b_{t-1}$ be the wires under the control of the adversary in top band and bottom band respectively.

Now suppose that the adversary controls $b_1, \ldots, b_{t-1}$ in such a way that none of these wires are pairwise consistent with the honest wire b_t. That is, vertex b_t becomes an isolated vertex in the consistency graph. As explained in Claim 11, the adversary can always control $b_1, \ldots, b_{t-1}$ in such a way which causes this situation. Moreover, let the adversary controls $b_1, \ldots, b_{t-1}$ in such a way that the consistency graph induced by the vertex set $\{b_1, \ldots, b_{t-1}\}$ results in a Tuŕan graph $T(t-1, \lceil (t-1)/3 \rceil)$. Again, since the wires $b_1, \ldots, b_{t-1}$ are under the control of the adversary, the adversary can control these wires so as to create the above situation. Now as stated in Theorem 3 and shown in Example 1, the graph $T(t-1, \lceil (t-1)/3 \rceil)$ will have $3^{t-1} = \mathcal{O}(3^t)$ maximal cliques. Moreover, from Claim 10, each of these maximal cliques will be considered as a distinct $\mathcal{B}_l$ by $\mathbf{S}$. Thus, $\mathbf{S}$ will get $\mathcal{O}(3^t)$ distinct $\mathcal{B}_l$'s at the end of **Phase II**. Next we show that the adversary can control $b_1, \ldots, b_{t-1}$ in such a way that each of these $\mathcal{O}(3^t)$ distinct $\mathcal{B}_l$'s become valid acceptable sets.

Recall that during **Phase II**, $\mathbf{R}$ sends $\beta^{\mathbf{R}}$ over the entire bottom band. Suppose that the adversary does not modify $\beta^{\mathbf{R}}$ during its transmission over wires

$b_1, \ldots, b_{t-1}$. In this case, all the honest wires in the top band, namely $f_1, \ldots, f_t$ will be added in each $\mathcal{F}_l$. Since each $\mathcal{B}_l$ contains at least one wire, it implies that $|\mathcal{F}_l| + |\mathcal{B}_l| \geq t+1$ will hold for all B_l's. Thus each $\mathcal{B}_l$ will be considered as acceptable. Thus there will be $\mathcal{O}(3^t)$ acceptable sets. □

Since **Phase III** of protocol $\Pi^{Existing}_{modified}$ is same as **Phase III** of protocol 3-SSMT, where **S** tried to send $m^{\mathbf{S}}$ using all possible acceptable sets, we have the following theorem:

Theorem 4. *In protocol* $\Pi^{Existing}_{modified}$, **S** *and* **R** *may have to do exponential computation. Moreover,* **S** *may do exponential communication.*

PROOF: As explained in previous lemma, **S** and **R** may end up with exponential number of acceptable sets in protocol $\Pi^{Existing}_{modified}$. Thus they have to do exponential computation. It is easy to see that in this case, **S** has to do exponential communication during **Phase III**, as **S** has to send $m^{\mathbf{S}}$, corresponding to each acceptable set by using the keys computed from it. □

Now as in protocol $\Pi^{Existing}_{modified}$, the adversary may control the bottom band in such a way that **S** may end up with exponential number of acceptable sets at the end of **Phase II** of the three phase efficient SSMT protocol $\Pi^{Existing}$ of [14, 55]. We capture this by the following theorem statement.

Theorem 5. *In protocol* $\Pi^{Existing}$, **S** *and* **R** *may have to do exponential computation. Moreover,* **S** *may do exponential communication.*

Since protocol $\Pi^{Existing}_{modified}$ is used as a black-box in other SRMT and SSMT protocols of [34], it will make the computational and communication complexity of all the SRMT and SSMT protocols of [34] exponential.

Remark 4 (Difference Between Phase II of Protocol 3-SSMT and $\Pi^{Existing}_{modified}$). *The main difference between* **Phase II** *of protocol 3-SSMT and* $\Pi^{Existing}_{modified}$ *is the way* **S** *divides the bottom band. In* $\Pi^{Existing}_{modified}$, *it is required that all the wires in a* $\mathcal{B}_l$ *should be pairwise consistent, thus making* $\mathcal{B}_l$ *a maximal clique. On the other hand, in 3-SSMT, it is required that all the wires in a* $\mathcal{B}_l$ *should be pairwise consistent only with wire* b_l. *It is this subtle difference which results in at most* u *acceptable sets (one corresponding to each wire in the bottom band) in protocol 3-SSMT.*

Though protocol 3-SSMT is an efficient SSMT protocol, it is not a communication optimal protocol. We can further reduce the communication complexity of SSMT protocols by increasing the number of phases in the protocol, which will further lead us to the design of a communication optimal SSMT protocol. In order to design the protocol, we require few more black box, which we describe in the subsequent sections.

Table 2: Matrix $B^{\mathbf{S}}$ as computed by $\mathbf{S}$ in Protocol 1-Pad

$k^{\mathbf{S}}_{1,1}$	$\ldots$	$k^{\mathbf{S}}_{1,c}$
$\ldots$	$\ldots$	$\ldots$
$k^{\mathbf{S}}_{i,1}$	$\ldots$	$k^{\mathbf{S}}_{i,c}$
$\ldots$	$\ldots$	$\ldots$
$k^{\mathbf{S}}_{t+1,1}$	$\ldots$	$k^{\mathbf{S}}_{t+1,c}$
$y^{\mathbf{S}}_{t+2,1}$	$\ldots$	$y^{\mathbf{S}}_{t+2,c}$
$\ldots$	$\ldots$	$\ldots$
$y^{\mathbf{S}}_{n,1}$	$\ldots$	$y^{\mathbf{S}}_{n,c}$

6. Six Phase Statistically Secure Pad Establishment Protocol

We now propose a six phase protocol called 6-Pad, which correctly establishes a random non-zero one time pad between $\mathbf{S}$ and $\mathbf{R}$ with very high probability by communicating $\mathcal{O}(n^3)$ field elements. Moreover, the pad will be information theoretically secure. If the entire bottom band is corrupted, then the size of the pad is $\Theta(n^2u)$ field elements. Otherwise the size of the pad is $\Theta(n^2)$ field elements. Before presenting protocol 6-Pad, we present another protocol called 1-Pad, which will be used as a black-box in protocol 6-Pad.

6.1. Single Phase Pad Establishment Protocol

Suppose $\mathbf{S}$ and $\mathbf{R}$ somehow in advance knows that full bottom band is corrupted. This implies that at most $t-u$ wires in the top band are corrupted. This further implies that there exists at least $t+1$ honest wires in the top band. Under this assumption, we design a single phase protocol called 1-Pad which allows $\mathbf{S}$ and $\mathbf{R}$ to correctly establish a non-zero random one time pad of size $\Theta(n^2u)$ with very high probability by communicating $\mathcal{O}(n^3)$ field elements. Moreover, the pad will be information theoretically secure. The idea of the protocol is similar to the one used in **Phase I** of protocol 3-SSMT. Recall that **Phase I** of protocol 3-SSMT would be successful if there exists at least $t+1$ honest wires in the top band (see Lemma 6). Now in 1-Pad we have $t+1$ honest wires in the top band. So if we execute **Phase I** of 3-SSMT then it would be successful. Protocol 1-Pad is based on this principle. The protocol is formally given in Fig. 6.

We now prove the properties of protocol 1-Pad.

Claim 12. *In protocol 1-Pad if* $\mathbf{R}$ *concludes that* $F_i^{\mathbf{R}}$ *is a valid row of* $B^{\mathbf{S}}$ *then with very high probability* $F_i^{\mathbf{R}} = F_i^{\mathbf{S}}$.

PROOF: The proof follows using similar argument as in Claim 1. □

Claim 13. *In protocol 1-Pad,* $\mathcal{P}^{\mathbf{S}}$ *will be information theoretically secure.*

Figure 6: Single Phase Protocol 1-Pad

Computation and Communication by S:

1. Let $c = n^2 + t - u$. $\mathbf{S}$ selects $(t+1) \times c$ random non-zero elements from $\mathbb{F}$, denoted by $k^{\mathbf{S}}_{1,1}, k^{\mathbf{S}}_{1,2}, \ldots, k^{\mathbf{S}}_{1,c}, \ldots, k^{\mathbf{S}}_{t+1,1}, k^{\mathbf{S}}_{t+1,2}, \ldots, k^{\mathbf{S}}_{t+1,c}$.
2. $\mathbf{S}$ constructs an $(t+1) \times c$ matrix $A^{\mathbf{S}}$, where the i^{th} row of $A^{\mathbf{S}}$ is $[k^{\mathbf{S}}_{i,1}, \ldots, k^{\mathbf{S}}_{i,c}]$, for $i = 1, \ldots, t+1$.
3. Considering the elements of i^{th} column $[k^{\mathbf{S}}_{1,i}, \ldots, k^{\mathbf{S}}_{t+1,i}]^T$, $\mathbf{S}$ forms a t degree polynomial $q_i(x)$ passing through the $t+1$ points $[(1, k^{\mathbf{S}}_{1,i}), (2, k^{\mathbf{S}}_{2,i}), \ldots, (t+1, k^{\mathbf{S}}_{t+1,i})]$, for $i = 1, \ldots, c$.
4. For $i = 1, \ldots, c$, $\mathbf{S}$ evaluates $q_i(x)$ at $x = t+2, \ldots, n$ to obtain $y^{\mathbf{S}}_{t+2,i}, \ldots, y^{\mathbf{S}}_{n,i}$ respectively.
5. Finally, $\mathbf{S}$ constructs the matrix $B^{\mathbf{S}}$ of size $n \times c$, where the i^{th} column of $B^{\mathbf{S}}$ is $[k^{\mathbf{S}}_{1,i}, \ldots, k^{\mathbf{S}}_{t+1,i}, y^{\mathbf{S}}_{t+2,i}, \ldots, y^{\mathbf{S}}_{n,i}]^T$, for $i = 1, \ldots, c$. The matrix $B^{\mathbf{S}}$ is pictorially shown in Table 2.
6. For $i = 1, \ldots, n$, $\mathbf{S}$ selects a random non-zero hash key $\alpha^{\mathbf{S}}_i$, corresponding to wire f_i.
7. For $i = 1, \ldots, n$, $\mathbf{S}$ sends the following to $\mathbf{R}$ over wire f_i:
   (a) The i^{th} row of $B^{\mathbf{S}}$, denoted by $F^{\mathbf{S}}_i$;
   (b) The hash key $\alpha^{\mathbf{S}}_i$ and
   (c) The hash values $v^{\mathbf{S}}_{ji}$, where $v^{\mathbf{S}}_{ji} = hash(\alpha^{\mathbf{S}}_i; F^{\mathbf{S}}_j)$, for $j = 1, \ldots, n$.
8. Let $\mathcal{V}^{\mathbf{S}}$ denote the concatenation of the elements of the first $t+1$ rows of $B^{\mathbf{S}}$. $\mathbf{S}$ computes
$$\mathcal{P}^{\mathbf{S}} = \mathbf{EXTRAND}_{|\mathcal{V}^{\mathbf{S}}|,(u+1)n^2}(\mathcal{V}^{\mathbf{S}}).$$
9. The vector $\mathcal{P}^{\mathbf{S}}$ denotes the information theoretically secure random pad of size $\Theta(n^2 u)$ which will be correctly established with $\mathbf{R}$ with very high probability.

Computation by R:

1. For $i = 1, \ldots, n$, let $\mathbf{R}$ receive the following over wire f_i:
   (a) The c-tuple, denoted by $F^{\mathbf{R}}_i$;
   (b) The hash key $\alpha^{\mathbf{R}}_i$ and
   (c) The hash values $v^{\mathbf{R}}_{ji}$, for $j = 1, \ldots, n$.
2. For $i = 1, \ldots, n$, $\mathbf{R}$ computes
$$Support_i = |\{j : hash(\alpha^{\mathbf{R}}_j; F^{\mathbf{R}}_i) = v^{\mathbf{R}}_{ij}\}|.$$
3. If $Support_i \geq t+1$, then $\mathbf{R}$ concludes that $F^{\mathbf{R}}_i$ is a valid row of $B^{\mathbf{S}}$. Otherwise, $\mathbf{R}$ concludes that $F^{\mathbf{R}}_i$ is an invalid row.
4. Using $t+1$ valid rows, $\mathbf{R}$ constructs the $n \times c$ array $B^{\mathbf{R}}$. From $B^{\mathbf{R}}$, $\mathbf{R}$ computes $\mathcal{V}^{\mathbf{R}}$, from which it finally computes $\mathcal{P}^{\mathbf{R}}$ and terminates..

PROOF: Recall that in protocol 1-Pad, there are at most $t-u$ wires in the top band which can be under the control of the adversary. Now using similar argument as in Claim 2, it follows that $[(t+1)-(t-u)]n^2 = (u+1)n^2$ elements of $\mathcal{V}^{\mathbf{S}}$ will be information theoretically secure. The rest now follows from the properties of **EXTRAND**. □

Claim 14. *If* $\mathbf{R}$ *gets* $t+1$ *valid rows of* $B^{\mathbf{S}}$ *then* $\mathbf{R}$ *can recover* $\mathcal{P}^{\mathbf{S}}$.

PROOF: Follows using similar argument as in Claim 3. □

Claim 15. *If* $\mathbf{R}$ *outputs* $\mathcal{P}^{\mathbf{R}}$ *then with very high probability* $\mathcal{P}^{\mathbf{R}} = \mathcal{P}^{\mathbf{S}}$.

PROOF: If $\mathbf{R}$ outputs $\mathcal{P}^{\mathbf{R}}$, then it implies that $\mathbf{R}$ has received $t+1$ valid rows. From Claim 12, all these rows are indeed the rows of $B^{\mathbf{S}}$ sent by $\mathbf{S}$ with very high probability. So from Claim 14, with very high probability, $\mathcal{P}^{\mathbf{R}} = \mathcal{P}^{\mathbf{S}}$. □

Theorem 6. *If the entire bottom band is corrupted, then protocol 1-Pad securely establishes a random non-zero pad of size* $\Theta(n^2u\kappa)$ *bits by communicating* $\mathcal{O}(n^3\kappa)$ *bits.*

PROOF: Over each wire, $\mathbf{S}$ sends $\mathcal{O}(n^2)$ field elements. This incurs a communication complexity of $\mathcal{O}(n^3)$ field elements and hence $\mathcal{O}(n^3\kappa)$ bits. The rest of the properties of protocol 1-Pad follows from Claim 12, Claim 13, Claim 14 and Claim 15. □

6.2. A Six Phase Pad Establishment Protocol

We now present out six phase pad establishment protocol 6-Pad. The main idea of the protocol is as follows: $\mathbf{S}$ and $\mathbf{R}$ interacts to find whether there the entire bottom band is corrupted or not. If they find that the entire bottom band is corrupted then $\mathbf{S}$ and $\mathbf{R}$ executes the single phase protocol 1-Pad to establish a pad of size $\Theta(n^2u)$ field elements. On the other hand if $\mathbf{S}$ and $\mathbf{R}$ finds that the entire bottom band is not corrupted then they apply **EXTRAND** on the information exchanged between them to establish a pad of size $\Theta(n^2)$ field elements.

Now as in the case of protocol 3-SSMT, we present protocol 6-Pad phase by phase. This will help the reader to understand the protocol conceptually. We begin with the description of the first two phases.

6.2.1. First Two Phases of Protocol 6-Pad

The first two phases of protocol 6-Pad are similar as in protocol 3-SSMT, except that now the first phase is initiated by $\mathbf{R}$. During first phase, $\mathbf{R}$ sends a random (n^2+1)-tuple to $\mathbf{S}$ over each wire. During second phase, $\mathbf{S}$ sends a random (n^2+t)-tuple to $\mathbf{R}$ over each wire. In addition to this, $\mathbf{S}$ and $\mathbf{R}$ also exchanges the hash values of the exchanged tuples, as in protocol 3-SSMT. The formal details of first two phases of 6-Pad are given in Fig. 7.

At the end of **Phase II** of protocol 6-Pad, $\mathbf{R}$ divides the top band using similar principle as used by $\mathbf{S}$ to divide the bottom band during protocol 3-SSMT. We now state the following lemmas, whose proofs are similar to the one given for protocol 3-SSMT.

Figure 7: First Two Phases of Protocol 6-Pad

Phase I: R to S:

1. For $i = 1, \ldots, u$, $\mathbf{R}$ selects a random non-zero $(n^2 + 1)$-tuple $(y^{\mathbf{R}}_{1,i}, \ldots, y^{\mathbf{R}}_{n^2+1,i})$, corresponding to wire b_i.
2. For $i = 1, \ldots, u$, $\mathbf{R}$ sends $(y^{\mathbf{R}}_{1,i}, \ldots, y^{\mathbf{R}}_{n^2+1,i})$ to $\mathbf{S}$ over wire b_i.

Phase II: S to R:

1. Let $\mathbf{S}$ receive $(y^{\mathbf{S}}_{1,i}, \ldots, y^{\mathbf{S}}_{n^2+1,i})$ along wire b_i, for $i = 1, \ldots, u$.
2. For $i = 1, \ldots, u$, $\mathbf{S}$ selects a random non-zero hash key $r^{\mathbf{S}}_i$, corresponding to wire b_i.
3. $\mathbf{S}$ computes the set $\beta^{\mathbf{S}} = \{(r^{\mathbf{S}}_i, \gamma^{\mathbf{S}}_i) : i = 1, \ldots, u\}$, where $\gamma^{\mathbf{S}}_i = hash(r^{\mathbf{S}}_i; y^{\mathbf{S}}_{1,i}, \ldots, y^{\mathbf{S}}_{n^2+1,i})$.
4. $\mathbf{S}$ associates a random non-zero (n^2+t)-tuple $(x^{\mathbf{S}}_{1,i}, \ldots, x^{\mathbf{S}}_{n^2+t,i})$ with wire f_i, for $i = 1, \ldots, n$.
5. For $i = 1, \ldots, n$, corresponding to wire f_i, $\mathbf{S}$ chooses n random, non-zero hash key $key^{\mathbf{S}}_{i,j}$, for $j = 1, \ldots, n$.
6. For $i = 1, \ldots, n$, $\mathbf{S}$ sends the following to $\mathbf{R}$ over wire f_i:
   (a) $\beta^{\mathbf{S}}$;
   (b) The $(n^2 + t)$-tuple $(x^{\mathbf{S}}_{1,i}, \ldots, x^{\mathbf{S}}_{n^2+t,i})$;
   (c) The pairs $\{(key^{\mathbf{S}}_{j,i}, \alpha^{\mathbf{S}}_{j,i}) : j = 1, \ldots, n\}$, where $\alpha^{\mathbf{S}}_{j,i} = hash(key^{\mathbf{S}}_{j,i}; x^{\mathbf{S}}_{1,j}, \ldots, x^{\mathbf{S}}_{n^2+t,j})$.

Computation by R at the end of Phase II:

1. Let $\mathbf{R}$ receive the following over wire f_i, for $i = 1, \ldots, n$:
   (a) $\beta^{\mathbf{S}}_i = \{(r^{\mathbf{R}}_{i,j}, \gamma^{\mathbf{R}}_{i,j}) : j = 1, \ldots, u\}$;
   (b) The $(n^2 + t)$-tuple $(x^{\mathbf{R}}_{1,i}, \ldots, x^{\mathbf{R}}_{n^2+t,i})$;
   (c) The pairs $\{(key^{\mathbf{R}}_{j,i}, \alpha^{\mathbf{R}}_{j,i}) : j = 1, \ldots, n\}$
2. For $i = 1, \ldots, n$, corresponding to wire f_i, $\mathbf{S}$ computes the set $\mathcal{F}_i$ and $\mathcal{B}_i$ as follows:
   (a) $\mathbf{R}$ adds wire $f_j \in \{f_1, \ldots, f_n\}$ to $\mathcal{F}_i$ (which is initially $\emptyset$) if f_i, f_j are found to be pairwise consistent by satisfying the following conditions:
      i. $\alpha^{\mathbf{R}}_{i,j} = hash(key^{\mathbf{R}}_{i,j}; x^{\mathbf{R}}_{1,i}, \ldots, x^{\mathbf{R}}_{n^2+t,i})$ and
      ii. $\alpha^{\mathbf{R}}_{j,i} = hash(key^{\mathbf{R}}_{j,i}; x^{\mathbf{R}}_{1,j}, \ldots, x^{\mathbf{R}}_{n^2+t,j})$.
   (b) $\mathbf{R}$ adds wire b_j to $\mathcal{B}_i$ (which is initially $\emptyset$) if $\gamma^{\mathbf{R}}_{i,j} = hash(r^{\mathbf{R}}_{i,j}; y^{\mathbf{R}}_{1,j}, \ldots, y^{\mathbf{R}}_{n^2+1,j})$.
   (c) If $|\mathcal{F}_i| + |\mathcal{B}_i| \geq t + 1$ then $\mathbf{R}$ considers $\mathcal{F}_i$ as an acceptable set. Otherwise $\mathbf{R}$ considers $\mathcal{F}_i$ as unacceptable.

Lemma 17. *Let f_i be an honest wire in the top band. Then $\mathcal{F}_i$ and corresponding $\mathcal{B}_i$ will have the following properties:*

1. *$\mathcal{F}_i$ will be an acceptable set.*
2. *All honest wires in the bottom band will be present in $\mathcal{B}_i$, while all the honest wires in the top band will be present in $\mathcal{F}_i$.*
3. *With very high probability, the (n^2+t)-tuple received by* **R** *at the end of* **Phase II** *over the wires in $\mathcal{F}_i$ are not modified.*
4. *With very high probability, the (n^2+1)-tuple received by* **S** *at the end of* **Phase I** *over the wires in $\mathcal{B}_i$ are not modified.*
5. *The adversary will have no information about the first n^2 values of the (n^2+t)-tuples which are exchanged over the honest wire(s) in $\mathcal{F}_i$. The adversary will also have no information about the first n^2 values of the (n^2+1)-tuples which are exchanged over the honest wires in $\mathcal{B}_i$.*

Lemma 18. *Let f_i be a wire in the top band such that f_i is under the control of the adversary. Moreover, let $\mathcal{F}_i$ be an acceptable set. Then $\mathcal{F}_i$ and corresponding $\mathcal{B}_i$ will have the following properties:*

1. *There will exist at least one honest wire, either from the top band or bottom band, which will be present in $\mathcal{F}_i$ or $\mathcal{B}_i$ respectively.*
2. *If there exists an honest wire in $\mathcal{F}_i$, then the (n^2+t)-tuple is correctly exchanged between* **S** *and* **R** *over that wire. Moreover, adversary will have no information about the first n^2 values of the (n^2+t)-tuple.*
3. *If there exists an honest wire in $\mathcal{B}_i$, then the (n^2+1)-tuple is correctly exchanged between* **S** *and* **R** *over that wire. Moreover, adversary will have no information about the first n^2 values of the (n^2+1)-tuple.*
4. *If there exists an honest wire in $\mathcal{F}_i$, then with very high probability the (n^2+t)-tuple is correctly exchanged between* **S** *and* **R** *over wire f_i. However, the adversary can modify the (n^2+t)-tuples which are exchanged over the corrupted wires in $\mathcal{F}_i$, other than f_i, without letting* **S** *and* **R** *know about it.*
5. *If there is no honest wire in $\mathcal{F}_i$, then the adversary can always modify the (n^2+t)-tuples which are exchanged over the corrupted wires in $\mathcal{F}_i$, without letting* **S** *and* **R** *know about it.*
6. *Irrespective of the number of honest wires in the bottom band, the adversary can always modify the (n^2+1)-tuple which is exchanged over a corrupted wire (if any) in $\mathcal{B}_i$, without letting* **S** *and* **R** *know about it.*

Lemma 19. *In protocol 6-Pad, there can be at most n acceptable sets.*

Lemma 20. *In protocol 6-Pad,* **R** *communicates $\mathcal{O}(n^2u)$ field elements during* **Phase I** *and* **S** *communicates $\mathcal{O}(n^3)$ field elements during* **Phase II**.

We now proceed to the description of the remaining phases of the protocol 6-Pad which are given in the next subsection.

6.2.2. Remaining Phases of Protocol 6-Pad

Corresponding to each acceptable set $\mathcal{F}_l$, $\mathbf{R}$ concatenates the tuples which are exchanged over the wires in $\mathcal{F}_l$ and $\mathcal{B}_l$. $\mathbf{R}$ then hashes the resultant tuple and sends the hash value, along with the identity of $\mathcal{F}_l$ and $\mathcal{B}_l$ to $\mathbf{S}$ through all the wires in $\mathcal{B}_l$. If the entire bottom band is corrupted, then $\mathbf{S}$ will not receive the correct hash value and with very high probability, $\mathbf{S}$ will come to know this. This is because there will exist at least one honest wire, either in $\mathcal{F}_l$ and $\mathcal{B}_l$, such that adversary will have no information about the first n^2 element of the tuple exchanged over the honest wire. In this case, $\mathbf{S}$ notifies about his finding to $\mathbf{R}$ and then executes protocol 1-Pad to establish a pad of size $\Theta(n^2u)$.

On the other hand, if there exists at least one honest wire in the bottom band, then $\mathbf{S}$ will correctly receive a valid hash value, along with the identity of corresponding $\mathcal{F}_l$ and $\mathcal{B}_l$. $\mathbf{S}$ then notifies $\mathbf{R}$ about the identity of $\mathcal{F}_l$ and $\mathcal{B}_l$. $\mathbf{S}$ also applies **EXTRAND** on the tuples which are exchanged along the wires in $\mathcal{F}_l$ and $\mathcal{B}_l$ to extract an information theoretically secure pad of size n^2. Application of **EXTRAND** is required because it may happen that $\mathcal{F}_l$ and $\mathcal{B}_l$ contains $t+1$ wires in total, out of which t are under the control of the adversary. So there will be only one honest wire and the adversary will not know the first n^2 elements of the tuple which is exchanged over that honest wire. Since $\mathbf{S}$ and $\mathbf{R}$ will not know the exact identity of the honest wire, they apply **EXTRAND**.

Notice that it is not easy for $\mathbf{S}$ to notify $\mathbf{R}$ about its finding. This is because there can be only $t+1$ wires in the top band and in the worst case, t of them can be corrupted. To deal with this problem, $\mathbf{S}$ notifies $\mathbf{R}$ about its finding using protocol 3-SSMT. Since 3-SSMT is an SSMT protocol, it will securely and hence correctly deliver the notification to $\mathbf{R}$. The formal details of the remaining phases of protocol 6-Pad are given in Fig. 8.

We now prove the properties of remaining phases of protocol 6-Pad, as given in Fig. 8.

Claim 16. *Let $\mathcal{F}_l$ be an acceptable set. Then adversary will have no information about n^2-1 elements of $\mathcal{V}_l^{\mathbf{R}}$.*

PROOF: Since $\mathcal{F}_l$ is an acceptable set, it implies that there exists at least one honest wire, either in $\mathcal{F}_l$ or corresponding set $\mathcal{B}_l$. Moreover, from Lemma 17 and Lemma 18, adversary will have no information about the first n^2 elements of the tuple which is exchanged over that honest wire. So $\mathcal{V}_l^{\mathbf{R}}$, which is the concatenation of the first n^2 elements from the tuples which are exchanged between $\mathbf{S}$ and $\mathbf{R}$ along the wires in $\mathcal{F}_l$ and $\mathcal{B}_l$ will have n^2 elements which will be unknown to the adversary. Now during **Phase III**, $\mathbf{R}$ sends one hash value of $\mathcal{V}_l^{\mathbf{R}}$ along the wires in $\mathcal{B}_l$. So from the properties of hashing, it still holds that n^2-1 elements of $\mathcal{V}_l^{\mathbf{R}}$ are information theoretically secure. □

Claim 17. *If at all $\mathbf{S}$ computes a pad $\mathcal{P}_1^{\mathbf{S}}$ of size n^2-1 field elements, then $\mathbf{R}$ will correctly output $\mathcal{P}_1^{\mathbf{R}} = \mathcal{P}_1^{\mathbf{S}}$, except with probability $2^{-\Omega(\kappa)}$.*

PROOF: Suppose $\mathbf{S}$ computes $\mathcal{P}_1^{\mathbf{S}}$ from $\mathcal{F}_{i,l}^{\mathbf{S}}$, $\mathcal{B}_{i,l}^{\mathbf{S}}$ and the tuple $(\mathcal{K}_{i,l}^{\mathbf{S}}, \delta_{i,l}^{\mathbf{S}})$. This implies that $\mathcal{V}_{i,l}^{\mathbf{S}}$ computed from the tuples exchanged over the wires in $\mathcal{F}_{i,l}^{\mathbf{S}}$

Figure 8: Remaining Phases of Protocol 6-Pad

Phase III: R to S: For each acceptable set $\mathcal{F}_l$ and the corresponding set $\mathcal{B}_l$, $\mathbf{R}$ does the following:

1. Let $\mathcal{V}_l^{\mathbf{R}}$ denote the concatenation of the first n^2 elements from (n^2+1)-tuples and (n^2+t)-tuples, which $\mathbf{R}$ has sent and received over the wires in $\mathcal{B}_l$ and $\mathcal{F}_l$ respectively.
2. Corresponding to vector $\mathcal{V}_l^{\mathbf{R}}$, $\mathbf{R}$ selects a random non-zero hash key $\mathcal{K}_l^{\mathbf{R}}$.
3. $\mathbf{R}$ computes $\delta_l^{\mathbf{R}} = hash(\mathcal{K}_l^{\mathbf{R}}; \mathcal{V}_l^{\mathbf{R}})$ and sends $\mathcal{B}_l, \mathcal{F}_l$ and the 2-tuple $(\mathcal{K}_l^{\mathbf{R}}, \delta_l^{\mathbf{R}})$ to $\mathbf{S}$ through all the wires in $\mathcal{B}_l$.

Computation by S at the end of Phase III: Now using the hash value(s) received from $\mathbf{R}$, $\mathbf{S}$ tries to find whether there exists at least one uncorrupted wire in the bottom band. For this, $\mathbf{S}$ does the following:

1. Let $\mathbf{S}$ receive the index set $\mathcal{F}_{i,l}^{\mathbf{S}}$ and $\mathcal{B}_{i,l}^{\mathbf{S}}$ and the 2-tuple $(\mathcal{K}_{i,l}^{\mathbf{S}}, \delta_{i,l}^{\mathbf{S}})$ along wire b_i, for $i = 1, \ldots, u$. Here $l \leq t+1$.
2. If for some $i \in \{1, \ldots, u\}$ and some $l \leq t+1$, $|\mathcal{F}_{i,l}^{\mathbf{S}}| + |\mathcal{B}_{i,l}^{\mathbf{S}}| \leq t$, then $\mathbf{S}$ concludes that wire b_i is corrupted and neglects all the values received along b_i.
3. If b_i is not neglected during previous step then for each $\mathcal{F}_{i,l}^{\mathbf{S}}$, $\mathcal{B}_{i,l}^{\mathbf{S}}$ and the tuple $(\mathcal{K}_{i,l}^{\mathbf{S}}, \delta_{i,l}^{\mathbf{S}})$ received along wire b_i, $\mathbf{S}$ does the following:
   (a) Let $\mathcal{V}_{i,l}^{\mathbf{S}}$ denote the concatenation of first n^2 values of the (n^2+1)-tuples and (n^2+t)-tuples, which $\mathbf{S}$ has received and sent over the wires in $\mathcal{B}_{i,l}^{\mathbf{S}}$ and $\mathcal{F}_{i,l}^{\mathbf{S}}$ respectively.
   (b) $\mathbf{S}$ now checks $\delta_{i,l}^{\mathbf{S}} \stackrel{?}{=} hash(\mathcal{K}_{i,l}^{\mathbf{S}}; \mathcal{V}_{i,l}^{\mathbf{S}})$.
   (c) If the test fails for all received $\mathcal{F}_{i,l}^{\mathbf{S}}$, $\mathcal{B}_{i,l}^{\mathbf{S}}$ and the tuple $(\mathcal{K}_{i,l}^{\mathbf{S}}, \delta_{i,l}^{\mathbf{S}})$ then $\mathbf{S}$ concludes that wire b_i is corrupted and neglects all the values received along b_i.
   (d) If the test succeeds for some $\mathcal{F}_{i,l}^{\mathbf{S}}$, $\mathcal{B}_{i,l}^{\mathbf{S}}$ and the tuple $(\mathcal{K}_{i,l}^{\mathbf{S}}, \delta_{i,l}^{\mathbf{S}})$ then then $\mathbf{S}$ does the following:
      i. $\mathbf{S}$ concludes that the tuples are correctly exchanged between $\mathbf{S}$ and $\mathbf{R}$ along the wires in $\mathcal{B}_{i,l}^{\mathbf{S}}$ and $\mathcal{F}_{i,l}^{\mathbf{S}}$.
      ii. $\mathbf{S}$ applies **EXTRAND** on $\mathcal{V}_{i,l}^{\mathbf{S}}$ to generate a vector $\mathcal{P}_1^{\mathbf{S}}$ of size $n^2 - 1$.
      iii. Finally $\mathbf{S}$ terminates 6-Pad by sending a special predefined "success" signal, along with the index of the wires in the set $\mathcal{B}_{i,l}^{\mathbf{S}}$ and $\mathcal{F}_{i,l}^{\mathbf{S}}$ to $\mathbf{R}$ by executing the protocol 3-SSMT.
      iv. At the end of 3-SSMT, $\mathbf{R}$ securely (and hence correctly) receives the set $\mathcal{B}_{i,l}^{\mathbf{S}}$ and $\mathcal{F}_{i,l}^{\mathbf{S}}$, computes $\mathcal{P}_1^{\mathbf{R}}$ and terminates 6-Pad. Since 3-SSMT takes three phases, $\mathbf{R}$ will terminate 6-Pad at the end of **Phase VI**.
4. If all the wires in the bottom band get discarded, then $\mathbf{S}$ concludes that entire bottom band is corrupted. In this case, $\mathbf{S}$ does the following:
   (a) $\mathbf{S}$ sends a special predefined "failure" signal to $\mathbf{R}$ by executing the three phase protocol 3-SSMT.
   (b) Parallely, $\mathbf{S}$ establishes a secure pad $\mathcal{P}_2^{\mathbf{S}}$ of size $\Theta(n^2 u)$ field elements with $\mathbf{R}$ by executing single phase Protocol 1-Pad.
   (c) At the end of 3-SSMT, $\mathbf{R}$ will know that the entire bottom band is corrupted.
   (d) Parallely at the end of 1-Pad, $\mathbf{R}$ will output the pad $\mathcal{P}_2^{\mathbf{R}}$ of size $\Theta(n^2 u)$ field elements. Since 3-SSMT takes three phases, $\mathbf{R}$ will terminate 6-Pad at the end of **Phase VI**.

and $\mathcal{B}^{\mathbf{S}}_{i,l}$ satisfies $\delta^{\mathbf{S}}_{i,l} = hash(\mathcal{K}^{\mathbf{S}}_{i,l}; \mathcal{V}^{\mathbf{S}}_{i,l})$. Now from the proof of the previous claim, $n^2 - 1$ elements of $\mathcal{V}^{\mathbf{S}}_{i,l}$ are information theoretically secure. So from the properties of hashing, it implies that the tuples are exchanged correctly between $\mathbf{S}$ and $\mathbf{R}$ along the wires in $\mathcal{F}^{\mathbf{S}}_{i,l}$ and $\mathcal{B}^{\mathbf{S}}_{i,l}$, except with probability $2^{-\Omega(\kappa)}$. Now $\mathbf{S}$ communicates the identity of the wires in $\mathcal{F}^{\mathbf{S}}_{i,l}$ and $\mathcal{B}^{\mathbf{S}}_{i,l}$ to $\mathbf{R}$ by executing protocol 3-SSMT. From the properties of 3-SSMT, $\mathbf{R}$ will correctly receive the identity of the wires in $\mathcal{F}^{\mathbf{S}}_{i,l}$ and $\mathcal{B}^{\mathbf{S}}_{i,l}$ except with probability $2^{-\Omega(\kappa)}$. So $\mathcal{P}^{\mathbf{R}}_1 = \mathcal{P}^{\mathbf{S}}_1$, except with probability $2^{-\Omega(\kappa)}$. □

Claim 18. *If at all* $\mathbf{S}$ *computes a pad* $\mathcal{P}^{\mathbf{S}}_2$ *of size* $\Theta(n^2u)$ *field elements, then* $\mathbf{R}$ *will correctly output* $\mathcal{P}^{\mathbf{R}}_2 = \mathcal{P}^{\mathbf{S}}_2$*, except with probability* $2^{-\Omega(\kappa)}$*.*

PROOF: The reason that $\mathbf{S}$ computes a pad $\mathcal{P}^{\mathbf{S}}_2$ of size $\Theta(n^2u)$ field elements is that all the wires in bottom band get rejected by $\mathbf{S}$ at the end of **Phase III**. This implies that $\mathbf{S}$ finds the entire bottom band to be corrupted, which $\mathbf{S}$ notifies to $\mathbf{R}$ by sending "failure" signal to $\mathbf{R}$ by executing protocol 3-SSMT. From the properties of 3-SSMT, $\mathbf{R}$ will correctly receive the "failure" signal and concludes that the entire bottom band is corrupted, except with error probability $2^{-\Omega(\kappa)}$. Now to establish the pad $\mathcal{P}^{\mathbf{S}}_2$, $\mathbf{S}$ executes the single phase protocol protocol 1-Pad. So from the properties of 1-Pad, $\mathbf{R}$ will correctly output $\mathcal{P}^{\mathbf{R}}_2 = \mathcal{P}^{\mathbf{S}}_2$ at the end of 1-Pad, except with probability $2^{-\Omega(\kappa)}$. □

Claim 19. *Irrespective of whether* $\mathbf{S}$ *and* $\mathbf{R}$ *agrees on a pad of size* $n^2 - 1$ *or* $\Theta(n^2u)$*, the pad will be information theoretically secure.*

PROOF: If $\mathbf{S}$ and $\mathbf{R}$ agrees on a pad of size $n^2 - 1$, it implies that the pad is computed by applying **EXTRAND** on some $\mathcal{V}^{\mathbf{S}}_{i,l}$. From the proof of Claim 17 and Claim 16, at least $n^2 - 1$ elements of $\mathcal{V}^{\mathbf{S}}_{i,l}$ are information theoretically secure. So from the properties of **EXTRAND**, the computed pad of size $n^2 - 1$ will be information theoretically secure.

On the other hand if the agreed pad is of size $\Theta(n^2u)$ then it implies that the pad is established by executing the protocol 1-Pad. In this case, security of the pad follows from the security of protocol 1-Pad. □

Claim 20. *In the steps given in Fig. 8,* $\mathbf{S}$ *and* $\mathbf{R}$ *has to communicate* $\mathcal{O}(n^3)$ *field elements.*

PROOF: During **Phase III**, corresponding to an acceptable set $\mathcal{F}_l$, $\mathbf{R}$ has to send the index of the wires in $\mathcal{F}_l, \mathcal{B}_l$ and the tuple $(\mathcal{K}^{\mathbf{R}}_l, \delta^{\mathbf{R}}_l)$ through all the wires in $\mathcal{B}_l$. This requires a communication complexity of $\mathcal{O}(nu)$ field elements. Since there can be n acceptable sets, it implies that $\mathbf{R}$ has to communicate $\mathcal{O}(n^2u) = \mathcal{O}(n^3)$ field elements during **Phase III**. During **Phase IV**, $\mathbf{S}$ will execute protocol 3-SSMT to notify either "success" or "failure" signal to $\mathbf{R}$. From Theorem 2, this requires a communication complexity of $\mathcal{O}(n^3)$ field elements. Thus $\mathbf{S}$ and $\mathbf{R}$ has to communicate $\mathcal{O}(n^3)$ field elements in the steps given in Fig. 8. □

Claim 21. *Protocol 6-Pad terminates in six phases.*

PROOF: Follows from the steps given in Fig. 7 and Fig. 8. □

Theorem 7 (Properties of Protocol 6-Pad). *Protocol 6-Pad has the following properties:*

1. *If the entire bottom band is corrupted, then* $\mathbf{S}$ *and* $\mathbf{R}$ *correctly establishes a pad of size* $\Theta(n^2u\kappa)$ *bits in six phases by communicating* $\mathcal{O}(n^3\kappa)$ *bits, except with error probability* $2^{-\Omega(\kappa)}$*. Moreover, the pad will be information theoretically secure.*
2. *If there exists at least one honest wire in the bottom band, then* $\mathbf{S}$ *and* $\mathbf{R}$ *correctly establishes a pad of size* $\Theta(n^2\kappa)$ *bits in six phases by communicating* $\mathcal{O}(n^3\kappa)$ *bits, except with error probability* $2^{-\Omega(\kappa)}$*. Moreover, the pad will be information theoretically secure.*

PROOF: Follows from Claim 16, Claim 17, Claim 18, Claim 19, Claim 20 and Claim 21. □

7. SRMT with Constant Factor Overhead

We now present an SRMT protocol called SRMT-Optimal which sends a message $m^{\mathbf{S}}$ containing ℓ field elements by communicating $\mathcal{O}(\ell)$ field elements with very high probability, where $\ell = (t - \frac{u}{2} + 1)n^2 = \Theta(n^3)$. The total communication complexity of the protocol is $\mathcal{O}(n^3)$ field elements and the protocol terminates in $\mathcal{O}(u)$ phases. Thus the protocol achieves reliability with constant factor overhead.

The idea behind the protocol is to create a win-win situation with the adversary as follows: if the adversary corrupts at most $t - \frac{u}{2}$ wires in the top band, then majority of the wires in the top band will be honest and $\mathbf{R}$ recovers the message from the information which it receives from the honest wires in the top band. On the other hand, if more than $t - \frac{u}{2}$ wires are corrupted in the top band, then majority wires in the bottom band will be honest and so both $\mathbf{S}$ and $\mathbf{R}$ comes to know about the identity of corrupted wires in the top band by using the honest wires in the bottom band. After knowing the identity of corrupted wires in the top band, $\mathbf{S}$ re-sends $m^{\mathbf{S}}$ so that $\mathbf{R}$ can recover it correctly.

As a part of pre-processing step, $\mathbf{S}$ and $\mathbf{R}$ securely establishes $\Theta(n^2)$ random non-zero elements from $\mathbb{F}$ with each other in advance with very high probability by executing the six phase protocol 6-Pad. This will require a communication complexity of $\mathcal{O}(n^3)$ field elements. Let the set of elements in the established pad be denoted by $\mathcal{K}$, which we call as **global key set**. The elements in $\mathcal{K}$ will be used by $\mathbf{S}$ and $\mathbf{R}$ as authentication and hash keys to reliably exchange the outcome of certain steps during the execution of the protocol SRMT-Optimal. Note that the elements in $\mathcal{K}$ need not be distinct, but they are randomly selected from $\mathbb{F}$. We assume that initially all the elements in $\mathcal{K}$ are marked as "unused". Each time $\mathbf{S}$ ($\mathbf{R}$) needs a key(s) for hashing or authentication, then

the first "unused" element(s) from $\mathcal{K}$ is/are selected as key(s). In order to do the verification, $\mathbf{R}$ ($\mathbf{S}$) also uses the same element(s) from $\mathcal{K}$ as keys. Once the verification is done, the element(s) is/are marked as "used" Thus we can view $\mathcal{K}$ as a global set, which is parallely used and updated by both $\mathbf{S}$ and $\mathbf{R}$. We now describe protocol SRMT-Optimal phase by phase. We begin with the description of first two phases, which is is given in the next section.

7.1. First Two Phases of Protocol SRMT-Optimal

Let $m^{\mathbf{S}} = [m^{\mathbf{S}}_{1,1}, \ldots, m^{\mathbf{S}}_{1,n^2}, m^{\mathbf{S}}_{2,1}, \ldots, m^{\mathbf{S}}_{2,n^2}, \ldots, m^{\mathbf{S}}_{t-\frac{u}{2}+1,1}, \ldots, m^{\mathbf{S}}_{t-\frac{u}{2}+1,n^2}]$ The first two phases of protocol SRMT-Optimal are given in Fig. 9.

We now prove the properties of first two phases of protocol SRMT-Optimal.

Claim 22. *Let f_i be a corrupted wire which has delivered $F^{\mathbf{R}}_i \neq F^{\mathbf{S}}_i$ to $\mathbf{R}$ and let f_j be an honest wire. Then with very high probability arc $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$.*

PROOF: Since f_j is honest, it correctly and hence securely delivers $\alpha^{\mathbf{R}}_j = \alpha^{\mathbf{S}}_j$ and $v^{\mathbf{R}}_{ij} = v^{\mathbf{S}}_{ij} = hash(\alpha^{\mathbf{S}}_j; F^{\mathbf{S}}_i)$ to $\mathbf{R}$. If $F^{\mathbf{R}}_i \neq F^{\mathbf{S}}_i$, then in order that $(f_i, f_j) \notin \mathcal{E}^{\mathbf{R}}$, $hash(\alpha^{\mathbf{S}}_j; F^{\mathbf{S}}_i) = hash(\alpha^{\mathbf{S}}_j; F^{\mathbf{R}}_i)$ should hold, even if $F^{\mathbf{R}}_i \neq F^{\mathbf{S}}_i$. But from the property of hashing, this can happen with probability at most $\frac{n^2-1}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$, which is negligible. □

Claim 23. *Let f_i be a corrupted wire which has delivered $F^{\mathbf{R}}_i \neq F^{\mathbf{S}}_i$ to $\mathbf{R}$. Then except with probability $2^{-\Omega(\kappa)}$, there will exist at least one arc $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$, such that wire f_j is honest.*

PROOF: The proof follows from the previous claim and the fact there exists at least one honest wire in the top band. □

Claim 24. *In protocol SRMT-Optimal, $\mathbf{S}$ communicates $\mathcal{O}(n^3)$ field elements during* **Phase I**, *while $\mathbf{R}$ communicates $\mathcal{O}(n^3)$ field elements during* **Phase II**.

PROOF: During **Phase I**, $\mathbf{S}$ communicates $\mathcal{O}(n^2)$ field elements over each wire. This incurs a total communication complexity of $\mathcal{O}(n^3)$ field elements. During **Phase II**, $\mathbf{R}$ sends the conflict list over the entire bottom band. In the worst case, the size of the conflict list may be $\mathcal{O}(n^2)$ field elements. So sending it over entire bottom band requires a communication cost of $\mathcal{O}(n^2 u) = \mathcal{O}(n^3)$ field elements, as $u = \mathcal{O}(n)$. □

Claim 25. *In protocol SRMT-Optimal, if there exists at most $t - \frac{u}{2}$ corrupted wires in the top band, then each wire $f_i \in \mathcal{P}^{\mathbf{R}}$ will deliver correct $F^{\mathbf{R}}_i = F^{\mathbf{S}}_i$ to $\mathbf{R}$, except with probability $2^{-\Omega(\kappa)}$.*

PROOF: Suppose there exists $t-\frac{u}{2}$ corrupted wires in the top band. Let $f_i \in \mathcal{P}^{\mathbf{R}}$. If f_i is honest then it implies that $F^{\mathbf{R}}_i = F^{\mathbf{S}}_i$. On the other hand let f_i be a corrupted wire, who has delivered $F^{\mathbf{R}}_i \neq F^{\mathbf{S}}_i$. Since $f_i \in \mathcal{P}^{\mathbf{R}}$, it implies that $|Support_i| \geq (t - \frac{u}{2} + 1)$, which further implies that there exists at least one honest wire, say f_j, such that $j \in Support_i$. This further implies that $v^{\mathbf{R}}_{ij} =$

Figure 9: First Two Phases of Protocol SRMT-Optimal

Phase I: S to R: $\mathbf{S}$ does the following computation and communication:

1. Using the elements of $m^{\mathbf{S}}$, $\mathbf{S}$ constructs an $(t - \frac{u}{2} + 1) \times n^2$ matrix $A^{\mathbf{S}}$, where the i^{th} row of $A^{\mathbf{S}}$ is $[m^{\mathbf{S}}_{i,1}, \ldots, m^{\mathbf{S}}_{i,n^2}]$, for $i = 1, \ldots, (t - \frac{u}{2} + 1)$.
2. $\mathbf{S}$ now constructs an $n \times n^2$ matrix $B^{\mathbf{S}}$ from $A^{\mathbf{S}}$ in same way as in protocol 1-Pad except that now $\mathbf{S}$ fits a polynomial of degree $(t - \frac{u}{2})$ passing through each column of $A^{\mathbf{S}}$.
3. For $i = 1, \ldots, n$, let $F^{\mathbf{S}}_i$ denote the i^{th} row of matrix $B^{\mathbf{S}}$.
4. For $i = 1, \ldots, n$, $\mathbf{S}$ selects a random, non-zero hash key $\alpha^{\mathbf{S}}_i$ corresponding to wire f_i.
5. For $i = 1, \ldots, n$, $\mathbf{S}$ sends the following to $\mathbf{R}$ over wire f_i:
   (a) The n^2-tuple $F^{\mathbf{S}}_i$;
   (b) The hash key $\alpha^{\mathbf{S}}_i$ and
   (c) The hash values $v^{\mathbf{S}}_{ji}$, where $v^{\mathbf{S}}_{ji} = hash(\alpha^{\mathbf{S}}_i; F^{\mathbf{S}}_j)$, for $j = 1, \ldots, n$.

Phase II: R to S: $\mathbf{R}$ does the following computation and communication:

1. For $i = 1, \ldots, n$, let $\mathbf{R}$ receive the following from $\mathbf{S}$ over wire f_i:
   (a) The n^2-tuple $F^{\mathbf{R}}_i$;
   (b) The hash key $\alpha^{\mathbf{R}}_i$ and
   (c) The hash values $v^{\mathbf{R}}_{ji}$, for $j = 1, \ldots, n$.
2. For $i = 1, \ldots, n$, $\mathbf{R}$ computes $Support_i = |\{j : v^{\mathbf{R}}_{ij} = hash(\alpha^{\mathbf{R}}_j; F^{\mathbf{R}}_i)\}|$.
3. Let $\mathcal{P}^{\mathbf{R}}$ denote the set of wires f_i, such that $Support_i \geq (t - \frac{u}{2} + 1)$.
4. $\mathbf{R}$ constructs a directed graph $\mathcal{G}^{\mathbf{R}} = (\mathcal{V}^{\mathbf{R}}, \mathcal{E}^{\mathbf{R}})$, called *conflict graph*, where $\mathcal{V}^{\mathbf{R}} = \{f_1, f_2, \ldots, f_n\}$ and arc $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$ if $v^{\mathbf{R}}_{ij} \neq hash(\alpha^{\mathbf{R}}_j; F^{\mathbf{R}}_i)$.
5. Corresponding to graph $\mathcal{G}^{\mathbf{R}}$, $\mathbf{R}$ constructs a conflict list $\mathcal{Y}^{\mathbf{R}}$ of five tuples where for each arc $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$, there exists a five tuple $(f_i, f_j, \alpha^{\mathbf{R}}_j, hash(\alpha^{\mathbf{R}}_j; F^{\mathbf{R}}_i), v^{\mathbf{R}}_{ij})$ in $\mathcal{Y}^{\mathbf{R}}$.
6. $\mathbf{R}$ sends $\mathcal{Y}^{\mathbf{R}}$ to $\mathbf{S}$ through all the wires in the bottom band.

$hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}})$. However, since f_j is an honest wire, it implies that $\alpha_j^{\mathbf{R}} = \alpha_j^{\mathbf{S}}$ and $v_{ij}^{\mathbf{R}} = v_{ij}^{\mathbf{S}}$, where $v_{ij}^{\mathbf{S}} = hash(\alpha_j^{\mathbf{S}}; F_i^{\mathbf{S}})$. Since the adversary does not know $\alpha_j^{\mathbf{S}}$, it follows from the properties of hashing that the adversary can ensure that $hash(\alpha_j^{\mathbf{S}}; F_i^{\mathbf{S}}) = hash(\alpha_j^{\mathbf{S}}; F_i^{\mathbf{R}})$, even if $F_i^{\mathbf{R}} = F_i^{\mathbf{S}}$, except with probability $\frac{n^2-1}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$. Thus except with probability $2^{-\Omega(\kappa)}$, $f_i \notin \mathcal{P}^{\mathbf{R}}$. □

Lemma 21. *In protocol SRMT-Optimal, if there exists at most $t - \frac{u}{2}$ corrupted wires in the top band, then except with error probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ can correctly recover $m^{\mathbf{R}} = m^{\mathbf{S}}$ by using the $F_i^{\mathbf{R}}$'s delivered by f_i's in $\mathcal{P}^{\mathbf{R}}$.*

Proof: If there exists at most $t - \frac{u}{2}$ corrupted wires in the top band then from the proof of Claim 25, each $f_i \in \mathcal{P}^{\mathbf{R}}$ has delivered correct $F_i^{\mathbf{R}} = F_i^{\mathbf{S}}$, except with error probability $2^{-\Omega(\kappa)}$. Moreover, there will be at least $t - \frac{u}{2} + 1$ wires in $\mathcal{P}^{\mathbf{R}}$, as all honest wires in the top band will be present in $\mathcal{P}^{\mathbf{R}}$. Now each $F_i^{\mathbf{S}}$ is a valid row of $B^{\mathbf{S}}$ and $t - \frac{u}{2} + 1$ valid rows are enough to construct the entire $B^{\mathbf{S}}$, as elements of each column of $B^{\mathbf{S}}$ are points on a $(t - \frac{u}{2})$ degree polynomial. So using the n^2-tuples delivered by the wires in $\mathcal{P}^{\mathbf{R}}$, $\mathbf{R}$ can reconstruct $B^{\mathbf{S}}$ and hence $A^{\mathbf{S}}$. Now the elements of $A^{\mathbf{S}}$ are nothing but the elements of $m^{\mathbf{S}}$. □

Lemma 21 shows that if some how $\mathbf{R}$ can find out whether there are at most $t - \frac{u}{2}$ corrupted wires in the top band, then $\mathbf{R}$ can recover $m^{\mathbf{S}}$ by using the n^2-tuples delivered by the wires in $\mathcal{P}^{\mathbf{R}}$. In order to find the status of the top band, $\mathbf{R}$ constructs the conflict list and sends it to $\mathbf{S}$. We now proceed to the description of **Phase III** of protocol SRMT-Optimal which is given in the next section.

7.2. Phase III and Phase IV of Protocol SRMT-Optimal

$\mathbf{S}$ waits for a conflict list, which is received identically through at least $\frac{u}{2} + 1$ wires. If $\mathbf{S}$ does not receive any conflict list identically through at least $\frac{u}{2} + 1$ wires, then $\mathbf{S}$ concludes that at least $\frac{u}{2} + 1$ wires are corrupted in the bottom band. This further implies that at most $t - \frac{u}{2} - 1$ wires are corrupted in the top band. In this case, the protocol proceeds as shown in Fig. 10.

The correctness of the protocol in this execution sequence is proved in Lemma 22.

Lemma 22. *If $\mathbf{S}$ does not receive the same conflict list through at least $\frac{u}{2} + 1$ wires in the bottom band then except with error probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ correctly recovers $m^{\mathbf{S}}$ from the n^2-tuples delivered by the wires in $\mathcal{P}^{\mathbf{R}}$. Moreover, in this case, SRMT-Optimal terminates in three phases. Furthermore, $\mathbf{S}$ will communicate $\mathcal{O}(n)$ field elements during* **Phase III**.

Proof: If $\mathbf{S}$ does not receive the same conflict list through at least $\frac{u}{2} + 1$ wires in the bottom band, then it implies that at least $\frac{u}{2} + 1$ wires in the bottom band are corrupted which further implies that at most $t - \frac{u}{2} - 1$ wires in the top band are corrupted. This further implies that each wire $f_i \in \mathcal{P}^{\mathbf{R}}$ has delivered correct $F_i^{\mathbf{R}} = F_i^{\mathbf{S}}$ to $\mathbf{R}$ during **Phase I**, except with probability $2^{-\Omega(\kappa)}$ (see Claim 25).

Figure 10: Execution of SRMT-Optimal If **S** Does Not Receive $\frac{u}{2}+1$ Identical Conflict Lists Through the Bottom Band

Phase III: S to R:

1. By selecting two elements from the global key set $\mathcal{K}$ as authentication keys, **S** authenticates an unique special predetermined signal "terminate1" and sends to **R** over the entire top band. Moreover, **S** terminates SRMT-Optimal.

Computation by R at the End of Phase III:

1. **R** correctly receives the "terminate1" signal with very high probability and concludes that at most $t-\frac{u}{2}$ wires have delivered incorrect values during **Phase I**.
2. By using the n^2-tuples received along the wires in $\mathcal{P}^{\mathbf{R}}$ during **Phase I**, **R** constructs the array $B^{\mathbf{R}}$. From $B^{\mathbf{R}}$, **R** recovers $m^{\mathbf{R}}$ and terminates SRMT-Optimal.

Moreover, from the proof of Lemma of 21, **R** can correctly recover $m^{\mathbf{R}}=m^{\mathbf{S}}$ by using the $F_i^{\mathbf{R}}$'s delivered by f_i's in $\mathcal{P}^{\mathbf{R}}$, except with error probability $2^{-\Omega(\kappa)}$. Since **S** authenticates the "terminate1" signal by using the keys from global key set $\mathcal{K}$, the keys will be unknown to the adversary. So except with error probability $2^{-\Omega(\kappa)}$, **R** will receive the "terminate1" signal and will conclude that at most $t-\frac{u}{2}-1$ wires in the top band are corrupted. Now as explained above, **R** will correctly recover $m^{\mathbf{S}}$ from the n^2-tuples delivered by the wires in $\mathcal{P}^{\mathbf{R}}$.

It is easy to see that in this case, protocol SRMT-Optimal terminates in three phases. Since **S** only sends the authentication of "terminate" signal over each wire during **Phase III**, it will require a communication cost of $\mathcal{O}(n)$ field elements. □

If **S** receives a unique conflict list $\mathcal{Y}^{\mathbf{S}}$ through at least $\frac{u}{2}+1$ wires in the bottom band then **S** cannot conclude anything about the status of the top band and bottom band. That is, **S** cannot determine whether the received conflict list is a genuine conflict list or not. **S** considers the received conflict list as genuine and from it, after doing local comparison, **S** tries to find the number of corrupted wires, which delivered incorrect $F_i^{\mathbf{S}}$'s to **R** during **Phase I**. **S** saves the identity of such wires in a list $L_{fault}^{\mathbf{S}}$. The steps performed by **S** to compute $L_{fault}^{\mathbf{S}}$ from $\mathcal{Y}^{\mathbf{S}}$ is shown in Fig. 11.

Before proceeding further, we make the following claims:

Claim 26. *Let majority of the wires in the bottom band are honest and let f_i be a corrupted wire in the top band who has delivered $F_i^{\mathbf{R}} \neq F_i^{\mathbf{S}}$ to* **R** *during*

Figure 11: Local Computation by S at the End of Phase II if **S** Receives $\frac{u}{2}+1$ Identical Conflict Lists Through the Bottom Band

Local Computation by S at the End of Phase II:

1. Let **S** receive the conflict list $\mathcal{Y}^{\mathbf{S}}$ through at least $\frac{u}{2}+1$ wires in the bottom band, where $\mathcal{Y}^{\mathbf{S}}$ is a collection of five-tuple of the form $(f_i, f_j, \alpha_j'^{\mathbf{S}}, \gamma_{ij}'^{\mathbf{S}}, v_{ij}'^{\mathbf{S}})$.
2. **S** creates a list $L_{fault}^{\mathbf{S}}$, which is initialized to $\emptyset$.
3. For each five-tuple $(f_i, f_j, \alpha_j'^{\mathbf{S}}, \gamma_{ij}'^{\mathbf{S}}, v_{ij}'^{\mathbf{S}})$ in $\mathcal{Y}^{\mathbf{S}}$, **S** does the following computation:
   (a) **S** checks $\alpha_j'^{\mathbf{S}} \stackrel{?}{=} \alpha_j^{\mathbf{S}}$ and $v_{ij}'^{\mathbf{S}} \stackrel{?}{=} v_{ij}^{\mathbf{S}}$.
   (b) If any of the above test fails then **S** concludes that wire f_j has delivered incorrect values to **R** during **Phase I** and adds f_j to a list $L_{fault}^{\mathbf{S}}$
   (c) If f_j is not added to $L_{fault}^{\mathbf{S}}$ then **S** checks $\gamma_{ij}'^{\mathbf{S}} \stackrel{?}{=} hash(\alpha_j^{\mathbf{S}}; F_i^{\mathbf{S}})$.
   (d) If the above test fails then **S** concludes that wire f_i has delivered incorrect $F_i^{\mathbf{R}} \neq F_i^{\mathbf{S}}$ to **R** during **Phase I** and adds f_i to $L_{fault}^{\mathbf{S}}$.

Phase I. *Then except with error probability* $2^{-\Omega(\kappa)}$, **S** *will include* f_i *in* $L_{fault}^{\mathbf{S}}$ *at the end of* **Phase II**.

PROOF: Let f_i be a corrupted wire in the top band who has delivered $F_i^{\mathbf{R}} \neq F_i^{\mathbf{S}}$ to **R** during **Phase I**. Then from Claim 23, except with error probability $2^{-\Omega(\kappa)}$, there will exist at least one arc (f_i, f_j) in the conflict graph, such that f_j is an honest wire. This further implies that the five tuple $(f_i, f_j, \alpha_j^{\mathbf{R}}, hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}}), v_{ij}^{\mathbf{R}})$ will be present in the conflict list $\mathcal{Y}^{\mathbf{R}}$, except with error probability $2^{-\Omega(\kappa)}$, such that $\alpha_j^{\mathbf{R}} = \alpha_j^{\mathbf{S}}$, $v_{ij}^{\mathbf{R}} = v_{ij}^{\mathbf{S}}$, but $hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}}) \neq v_{ij}^{\mathbf{R}}$. Now if the majority of the wires in the bottom band are honest then **S** will correctly receive $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$ through the majority wires in the bottom band. This implies that **S** will correctly receive $(f_i, f_j, \alpha_j'^{\mathbf{S}}, \gamma_{ij}'^{\mathbf{S}}, v_{ij}'^{\mathbf{S}}) = (f_i, f_j, \alpha_j^{\mathbf{R}}, hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}}), v_{ij}^{\mathbf{R}})$. So after doing the local comparison, **S** will find that $\alpha_j'^{\mathbf{S}} = \alpha_j^{\mathbf{S}}$ and $v_{ij}'^{\mathbf{S}} = v_{ij}^{\mathbf{S}}$ and hence f_j will not be added in $L_{fault}^{\mathbf{S}}$. But **S** will find that $\gamma_{ij}'^{\mathbf{S}} \neq hash(\alpha_j^{\mathbf{S}}; F_j^{\mathbf{S}})$. So **S** will add wire f_i in $L_{fault}^{\mathbf{S}}$. □

Claim 27. *Let majority of the wires in the bottom band are honest. Then* **S** *will not include any honest* f_i *in* $L_{fault}^{\mathbf{S}}$ *at the end of* **Phase II**.

PROOF: First of all notice that if at all a five tuple $(f_i, f_j, \alpha_j^{\mathbf{R}}, hash(\alpha_j^{\mathbf{R}}; F_i^{\mathbf{R}}), v_{ij}^{\mathbf{R}})$ is present in the conflict list $\mathcal{Y}^{\mathbf{R}}$, then at least one of the wires f_i or f_j is corrupted. This is because no two honest wires will conflict each other. Now suppose f_i is the corrupted wire. If majority of the wires in the bottom band

are honest then $\mathbf{S}$ will correctly receive $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$ through the majority wires in the bottom band. Now as shown in the previous claim, after doing local comparison, $\mathbf{S}$ will find out that wire f_i is corrupted and includes only f_i in $L^{\mathbf{S}}_{fault}$. $\square$

After finding $L^{\mathbf{S}}_{fault}$, $\mathbf{S}$ proceeds as follows: If $L^{\mathbf{S}}_{fault}$ contains at most $t-\frac{u}{2}$ wires then $\mathbf{S}$ can conclude that $\mathbf{R}$ has received sufficient information during **Phase I** to recover $m^{\mathbf{S}}$. This is because if at all $\mathcal{Y}^{\mathbf{S}}$ is a valid conflict list, which was indeed sent by $\mathbf{R}$ then the wires in $L^{\mathbf{S}}_{fault}$ are indeed genuinely corrupted, who delivered incorrect $F_i^{\mathbf{S}}$'s to $\mathbf{R}$ during **Phase I** (see Claim 26) and there are at most $t-\frac{u}{2}$ such $F_i^{\mathbf{S}}$'s. This implies that the remaining $t-\frac{u}{2}+1$ wires have delivered correct $F_i^{\mathbf{S}}$'s, which are sufficient to reconstruct the matrix $B^{\mathbf{S}}$ and hence the message $m^{\mathbf{S}}$.

On the other hand, if $\mathcal{Y}^{\mathbf{S}}$ is not a valid conflict list, then it implies that majority wires in the bottom band are corrupted, which further implies that majority wires in the top band are honest. So the wires in $\mathcal{P}^{\mathbf{R}}$ have delivered correct $F_i^{\mathbf{S}}$'s during **Phase I** with very high probability. So the main goal of $\mathbf{S}$ will be to some how reliably send back the received conflict list $\mathcal{Y}^{\mathbf{S}}$ and the corresponding list $L^{\mathbf{S}}_{fault}$, so that $\mathbf{R}$ can find out whether $\mathbf{S}$ has correctly received the original conflict list $\mathcal{Y}^{\mathbf{R}}$ through the majority wires in the bottom band. This is what $\mathbf{S}$ does during **Phase III**, as shown in Fig. 12. Notice that Fig. 12 represents the execution sequence when $\mathbf{S}$ finds that there are at most $t-\frac{u}{2}$ wires in the list $L^{\mathbf{S}}_{fault}$. The execution sequence when $L^{\mathbf{S}}_{fault}$ contains more than $t-\frac{u}{2}$ wires will be discussed after wards.

We now prove the properties of protocol SRMT-Optimal, if the protocol follows the steps given in Fig. 12.

Lemma 23. *If $\mathbf{S}$ receives the same conflict list through at least $\frac{u}{2}+1$ wires in the bottom band and if the size of resultant $L^{\mathbf{S}}_{fault}$ is at most $(t-\frac{u}{2})$ then except with error probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ will correctly output $m^{\mathbf{R}} = m^{\mathbf{S}}$ at the end of* **Phase III**. *In this case, protocol SRMT-Optimal will terminate in three phases and $\mathbf{S}$ will communicate $\mathcal{O}(n^3)$ field elements during* **Phase III**.

PROOF: Let $\mathbf{S}$ receive the conflict graph $\mathcal{Y}^{\mathbf{S}}$ through at least $\frac{u}{2}+1$ wires in the bottom band and let $L^{\mathbf{S}}_{fault}$ be the corresponding list computed by $\mathbf{S}$ from $\mathcal{Y}^{\mathbf{S}}$. Moreover, let $|L^{\mathbf{S}}_{fault}| \leq (t-\frac{u}{2})$. Now there are two possibilities:

1. $\underline{\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}}$: In this case, except with probability $2^{-\Omega(\kappa)}$, $\mathbf{S}$ will find out all corrupted f_i's, who have delivered $F_i^{\mathbf{R}} \neq F_i^{\mathbf{S}}$ during **Phase I** and includes them in $L^{\mathbf{S}}_{fault}$ (see Claim 26). Moreover, from Claim 27, only corrupted f_i's will be included in $L^{\mathbf{S}}_{fault}$. Since $|L^{\mathbf{S}}_{fault}| \leq (t-\frac{u}{2})$, it implies that at least $(t-\frac{u}{2})+1$ f_i's delivered correct $F_i^{\mathbf{S}}$'s during **Phase I**. Now during **Phase III**, $\mathbf{S}$ sends the received $\mathcal{Y}^{\mathbf{S}}$, corresponding $L^{\mathbf{S}}_{fault}$, authentication of $\mathcal{Y}^{\mathbf{S}}$ and the authentication of $L^{\mathbf{S}}_{fault}$ through the entire top band. Notice that the authentication keys used to authenticate $\mathcal{Y}^{\mathbf{S}}$ and $L^{\mathbf{S}}_{fault}$ belong to $\mathcal{K}$ and so adversary does not have any information

Figure 12: Execution of SRMT-Optimal If $\mathbf{S}$ Receives $\frac{u}{2}+1$ Identical Conflict Lists and $|L^{\mathbf{S}}_{fault}| \leq (t-\frac{u}{2})$

Phase III: S to R:

1. By selecting two keys from the global key set $\mathcal{K}$, $\mathbf{S}$ authenticates special "terminate2" signal using $URauth$ function and sends it to $\mathbf{R}$ through the top band.
2. By selecting $2|L^{\mathbf{S}}_{fault}|$ keys and $10|\mathcal{Y}^{\mathbf{S}}|$ keys from global key set $\mathcal{K}$, $\mathbf{S}$ authenticates each element of $L^{\mathbf{S}}_{fault}$ and $\mathcal{Y}^{\mathbf{S}}$ respectively using $URauth$ function. Let $L^{\mathbf{S}}_{fault_{auth}}$ and $\mathcal{Y}^{\mathbf{S}}_{auth}$ denote the set of corresponding authenticated values.
3. $\mathbf{S}$ then sends $(\mathcal{Y}^{\mathbf{S}}, L^{\mathbf{S}}_{fault}, \mathcal{Y}^{\mathbf{S}}_{auth}, L^{\mathbf{S}}_{fault_{auth}})$ to $\mathbf{R}$ through the top band and terminates the protocol SRMT-Optimal.

Computation by R at the End of Phase III:

1. With very high probability, $\mathbf{R}$ correctly receives "terminate2" signal.
2. Let $\mathbf{R}$ receive $(\mathcal{Y}^{\mathbf{R}}_i, L^{\mathbf{R}}_{fault_i}, \mathcal{Y}^{\mathbf{R}}_{i,auth}, L^{\mathbf{R}}_{fault_{i,auth}})$ from $\mathbf{S}$ along wire f_i, for $i = 1, \ldots, n$. From these values, $\mathbf{R}$ now tries to find out whether $\mathbf{S}$ has correctly received the original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2}+1$ wires during **Phase I**, and if yes, then the corresponding $L^{\mathbf{S}}_{fault}$. For this, $\mathbf{R}$ does the following computation:
   (a) For each $i = 1, \ldots, n$, $\mathbf{R}$ checks $\mathcal{Y}^{\mathbf{R}}_i \stackrel{?}{=} \mathcal{Y}^{\mathbf{R}}$ and $|L^{\mathbf{R}}_{fault_i}| \leq (t-\frac{u}{2})$. If any of these test fails, then $\mathbf{R}$ neglects all the values received along f_i. Otherwise, $\mathbf{R}$ applies the $URauth$ function to each element of $\mathcal{Y}^{\mathbf{R}}_i$ by using the same keys from $\mathcal{K}$, which were used by $\mathbf{S}$ to authenticate $\mathcal{Y}^{\mathbf{S}}$ and computes the set $\mathcal{Y}'^{\mathbf{R}}_{i,auth}$. Similarly, $\mathbf{R}$ applies $URauth$ function to each element of $L^{\mathbf{R}}_{fault_i}$ by using the same keys from $\mathcal{K}$, which were used by $\mathbf{S}$ to authenticate $L^{\mathbf{S}}_{fault}$ and computes the set $L'^{\mathbf{R}}_{fault_{i,auth}}$. $\mathbf{R}$ then checks $\mathcal{Y}'^{\mathbf{R}}_{i,auth} \stackrel{?}{=} \mathcal{Y}^{\mathbf{R}}_{i,auth}$ and $L'^{\mathbf{R}}_{fault_{i,auth}} \stackrel{?}{=} L^{\mathbf{R}}_{fault_{i,auth}}$. If the test fails then $\mathbf{R}$ discards the values received along f_i.
   (b) If all the wires in the top band are discarded by $\mathbf{R}$ during previous step, then $\mathbf{R}$ concludes that $\mathbf{S}$ has not received original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2}+1$ wires during **Phase II**, which further implies that at most $t-\frac{u}{2}-1$ wires were corrupted in the top band during **Phase I**. So $\mathbf{R}$ recovers $m^{\mathbf{R}}$ by using the $F^{\mathbf{R}}_i$'s received over the wires in $\mathcal{P}^{\mathbf{R}}$ during **Phase I** and terminates SRMT-Optimal.
   (c) If there exists an $i \in \{1, 2, \ldots, n\}$ such that $\mathcal{Y}^{\mathbf{R}}_i = \mathcal{Y}^{\mathbf{R}}$, $|L^{\mathbf{R}}_{fault_i}| \leq (t-\frac{u}{2})$, $\mathcal{Y}'^{\mathbf{R}}_{i,auth} = \mathcal{Y}^{\mathbf{R}}_{i,auth}$ and $L'^{\mathbf{R}}_{fault_{i,auth}} = L^{\mathbf{R}}_{fault_{i,auth}}$, then $\mathbf{R}$ concludes that $\mathbf{S}$ has correctly received original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2}+1$ wires during **Phase II** and $L^{\mathbf{R}}_{fault_i}$ is the corresponding L_{fault} sent by $\mathbf{S}$. Now by using the $F^{\mathbf{R}}_i$'s received over the wires not in $L^{\mathbf{R}}_{fault_i}$, $\mathbf{R}$ recovers $m^{\mathbf{R}}$ and terminates SRMT-Optimal.

about them. Moreover, there exists at least one honest wire in the top band. So from the properties of $URauth$, $\mathbf{R}$ will correctly receive $L^{\mathbf{S}}_{fault}$ with very high probability. By neglecting the wires in $L^{\mathbf{S}}_{fault}$, $\mathbf{R}$ will be left with at least $(t - \frac{u}{2} + 1)$ wires in the top band and each of them has delivered correct $F^{\mathbf{S}}_i$ with very high probability. Now it is easy to see that using these $F^{\mathbf{S}}_i$'s, $\mathbf{R}$ will correctly recover $B^{\mathbf{S}}$ and hence $m^{\mathbf{S}}$. It is easy to see that in this case protocol SRMT-Optimal terminates at the end of **Phase III**.

2. $\underline{\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}}$: This implies that at least $\frac{u}{2} + 1$ wires in the bottom band are corrupted, which further implies that at most $t - \frac{u}{2} - 1$ wires are corrupted in the top band. So from Claim 25, all wires in $\mathcal{P}^{\mathbf{R}}$ have delivered correct $F^{\mathbf{R}}_i$'s during **Phase I** with very high probability. Moreover, if somehow $\mathbf{R}$ comes to know that there are at most $t - \frac{u}{2} - 1$ corrupted wires in the top band, then from the proof of Lemma 21, $\mathbf{R}$ can recover $m^{\mathbf{S}}$ correctly with very high probability by using the $F^{\mathbf{R}}_i$'s delivered by the wires in $\mathcal{P}^{\mathbf{R}}$. We now show that at the end of **Phase III**, with very high probability, $\mathbf{R}$ will come to know that at most $t - \frac{u}{2} - 1$ wires are corrupted in the top band.

   Note that $\mathbf{S}$ compute $L^{\mathbf{S}}_{fault}$ by doing local comparisons on the values in $\mathcal{Y}^{\mathbf{S}}$. Also $\mathcal{Y}^{\mathbf{S}}_{auth}$ and $L^{\mathbf{S}}_{fault_{auth}}$ are obtained by applying $URauth$ function to the elements of $\mathcal{Y}^{\mathbf{S}}$ and $L^{\mathbf{S}}_{fault}$ respectively, where the keys for authentication are selected from secret, global key set $\mathcal{K}$. So adversary will have no information about the authentication keys used for authenticating $\mathcal{Y}^{\mathbf{S}}$ and $L^{\mathbf{S}}_{fault}$. During **Phase III**, $\mathbf{S}$ sends $(\mathcal{Y}^{\mathbf{S}}, L^{\mathbf{S}}_{fault}, \mathcal{Y}^{\mathbf{S}}_{auth}, L^{\mathbf{S}}_{fault_{auth}})$ to $\mathbf{R}$ through the top band.

   Now suppose some wire f_i delivers $(\mathcal{Y}^{\mathbf{R}}_i, L^{\mathbf{R}}_{fault_i}, \mathcal{Y}^{\mathbf{R}}_{i,auth}, L^{\mathbf{R}}_{fault_{i,auth}})$. If f_i is honest, then $\mathcal{Y}^{\mathbf{R}}_i = \mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$. So $\mathbf{R}$ will neglect wire f_i. On the other hand if f_i is corrupted, then the adversary can ensure that $\mathcal{Y}^{\mathbf{R}}_i \neq \mathcal{Y}^{\mathbf{S}}$ but $\mathcal{Y}^{\mathbf{R}}_i = \mathcal{Y}^{\mathbf{R}}$. However, from the properties of $URauth$, without knowing the authentication keys used by $\mathbf{S}$ for authenticating $\mathcal{Y}^{\mathbf{S}}$, the adversary cannot produce the authentication of $\mathcal{Y}^{\mathbf{R}}_i = \mathcal{Y}^{\mathbf{R}}$, except with probability $2^{-\Omega(\kappa)}$. So except with error probability $2^{-\Omega(\kappa)}$, wire f_i will be caught and hence will be discarded by $\mathbf{R}$. This further implies that except with error probability all wires in top band will be discarded by $\mathbf{R}$ and hence $\mathbf{R}$ will conclude that $\mathbf{S}$ has not received $\mathcal{Y}^{\mathbf{R}}$ through at least $\frac{u}{2} + 1$ wires in the bottom band. Now as explained above, $\mathbf{R}$ can recover $m^{\mathbf{S}}$ correctly with very high probability by using the $F^{\mathbf{R}}_i$'s delivered by the wires in $\mathcal{P}^{\mathbf{R}}$. It is easy to see that in this case protocol SRMT-Optimal terminates at the end of **Phase III**.

Notice that if $\mathbf{S}$ executes **Phase III** as given in Fig. 12, then $\mathbf{S}$ sends $(\mathcal{Y}^{\mathbf{S}}, L^{\mathbf{S}}_{fault}, \mathcal{Y}^{\mathbf{S}}_{auth}, L^{\mathbf{S}}_{fault_{auth}})$ through entire top band. It is easy to see that this requires a communication complexity of $\mathcal{O}(n^3)$ field elements, as $|\mathcal{Y}^{\mathbf{S}}| = \mathcal{O}(n^2)$. □

Now we draw our attention to the execution of SRMT-Optimal when $\mathbf{S}$ receive $\mathcal{Y}^{\mathbf{S}}$ through $\frac{u}{2} + 1$ wires in the bottom band and the resultant $L^{\mathbf{S}}_{fault}$ has more

than $(t - \frac{u}{2})$ wires. In this case, $\mathbf{S}$ cannot say anything about the status of top band. To find whether indeed $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$, $\mathbf{S}$ authenticates $\mathcal{Y}^{\mathbf{S}}$ and $L^{\mathbf{S}}_{fault}$, sends it to $\mathbf{R}$ and wait for the feedback. If indeed $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$ then the wires in $L^{\mathbf{S}}_{fault}$ are indeed corrupted and so $\mathbf{S}$ and $\mathbf{R}$ ignores them and further continue with the protocol. The steps for further computation and communication will be discusses later. However, if $\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$ then with very high probability, $\mathbf{R}$ will detect this. In this case, $\mathbf{R}$ can easily recover $m^{\mathbf{S}}$ by using the $F_i^{\mathbf{R}}$'s delivered by the wires in $\mathcal{P}^{\mathbf{R}}$ during **Phase I**. So $\mathbf{R}$ recovers $m^{\mathbf{S}}$ and asks $\mathbf{S}$ to terminate the protocol. The formal details are presented in Fig. 13.

We now prove the properties of protocol SRMT-Optimal, if the protocol follows the steps given in Fig. 13.

Lemma 24. *If $\mathbf{S}$ receives the same conflict list $\mathcal{Y}^{\mathbf{S}}$ through at least $\frac{u}{2}+1$ wires in the bottom band and if the size of resultant $L^{\mathbf{S}}_{fault}$ is more than $(t - \frac{u}{2} + 1)$ then:*

1. *If $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$ then both $\mathbf{R}$ and $\mathbf{S}$ will come to know about this at the end of* **Phase III** *and* **Phase IV** *respectively. Moreover, both $\mathbf{S}$ and $\mathbf{R}$ will come to know the identity of $|L^{\mathbf{S}}_{fault}| \geq (t - \frac{u}{2} + 1)$ corrupted wires in the top band. Furthermore, both $\mathbf{S}$ and $\mathbf{R}$ will know that the majority of the wires in the bottom band are honest.*
2. *If $\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$ then except with probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ will come to know this at the end of* **Phase III**. *Moreover, $\mathbf{R}$ will recover $m^{\mathbf{R}}$ at the end of* **Phase III** *by using the $F_i^{\mathbf{R}}$'s delivered by the wires in $\mathcal{P}^{\mathbf{R}}$. Furthermore, $\mathbf{S}$ will terminate the protocol at the end of* **Phase IV**.
3. *During* **Phase III**, *$\mathbf{S}$ will communicate $\mathcal{O}(n^3)$ field elements while during* **Phase IV**, *$\mathbf{R}$ will communicate $\mathcal{O}(u)$ field elements.*

PROOF: Let $\mathbf{S}$ receive the same conflict list $\mathcal{Y}^{\mathbf{S}}$ through at least $\frac{u}{2}+1$ wires in the bottom band at the end of **Phase II**. Moreover, let the resultant $L^{\mathbf{S}}_{fault}$ computed from $\mathcal{Y}^{\mathbf{S}}$ be of size more than $(t - \frac{u}{2} + 1)$. Now there are two possible cases:

1. $\underline{\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}}$: This case is similar to the case $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$ in the proof of Lemma 23. Specifically, at the end of **Phase III**, $\mathbf{R}$ will conclude that $\mathbf{S}$ has correctly received $\mathcal{Y}^{\mathbf{R}}$ over the majority wires in the bottom band during **Phase II**. Moreover, except with error probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ will correctly receive $L^{\mathbf{S}}_{fault}$. Furthermore, each wire in $L^{\mathbf{S}}_{fault}$ will be indeed corrupted. So $\mathbf{R}$ will remove the wires in $L^{\mathbf{S}}_{fault}$ from his consideration for all future computation and communication. Since $|L_{fault}| \geq (t - \frac{u}{2} + 1)$, this implies that in the bottom band, there will be at most $\frac{u}{2} - 1$ corrupted wires and hence $\mathbf{R}$ concludes that the majority of the wires in the bottom band are honest. Since in this case $\mathbf{R}$ authenticates "continue" signal and sends to $\mathbf{S}$, at the end of end of **Phase IV**, $\mathbf{S}$ will correctly receive the signal over at least $\frac{u}{2}+1$ wires in the bottom band and thus will conclude that $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$. So $\mathbf{S}$ will also remove the wires in $L^{\mathbf{S}}_{fault}$ from

Figure 13: Execution of SRMT-Optimal If $\mathbf{S}$ Receives $\frac{u}{2}+1$ Identical Conflict Lists and $|L^{\mathbf{S}}_{fault}| \geq (t-\frac{u}{2})+1$

Phase III: S to R: Same as in Fig. 12, except that $\mathbf{S}$ does not send "terminate2" signal. Moreover, here $|L^{\mathbf{S}}_{fault}| \geq (t-\frac{u}{2})+1$.

Phase IV: R to S:

1. Let $\mathbf{R}$ receive $(\mathcal{Y}^{\mathbf{R}}_i, L^{\mathbf{R}}_{fault_i}, \mathcal{Y}^{\mathbf{R}}_{i,auth}, L^{\mathbf{R}}_{fault_{i,auth}})$ from $\mathbf{S}$ along wire f_i, for $i = 1, \ldots, n$. From these values, $\mathbf{R}$ tries to find out whether $\mathbf{S}$ has correctly received the original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2}+1$ wires during **Phase I**, and if yes, then the corresponding $L^{\mathbf{S}}_{fault}$. For this, $\mathbf{R}$ does the same computation as done by $\mathbf{R}$ at the end of **Phase III** in Fig. 12. However, now instead of checking $|L^{\mathbf{R}}_{fault_i}| \leq (t-\frac{u}{2})$, $\mathbf{R}$ checks $|L^{\mathbf{R}}_{fault_i}| \geq (t-\frac{u}{2}+1)$.
2. If all the wires in the top band are discarded by $\mathbf{R}$ then $\mathbf{R}$ concludes that $\mathbf{S}$ has not received original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2}+1$ wires during **Phase II**, which further implies that at most $t-\frac{u}{2}-1$ wires were corrupted in the top band during **Phase I**. So $\mathbf{R}$ recovers $m^{\mathbf{R}}$ by using the $F^{\mathbf{R}}_i$'s received over the wires in $\mathcal{P}^{\mathbf{R}}$ during **Phase I**. Moreover, $\mathbf{R}$ computes $response_1 = URauth(\text{"}terminate\text{"}; k_1, k_2)$, where k_1, k_2 are selected from $\mathcal{K}$. Finally $\mathbf{R}$ asks $\mathbf{S}$ to terminate SRMT-Optimal by sending $(\text{"}terminate\text{"}, response_1)$ over the bottom band and terminates SRMT-Optimal.
3. If $\mathbf{R}$ finds an $i \in \{1, 2, \ldots, n\}$ such that $\mathcal{Y}^{\mathbf{R}}_i = \mathcal{Y}^{\mathbf{R}}$, $|L^{\mathbf{R}}_{fault_i}| \geq (t-\frac{u}{2}+1)$, $\mathcal{Y}^{'\mathbf{R}}_{i,auth} = \mathcal{Y}^{\mathbf{R}}_{i,auth}$ and $L^{'\mathbf{R}}_{fault_{i,auth}} = L^{\mathbf{R}}_{fault_{i,auth}}$, then $\mathbf{R}$ concludes that $\mathbf{S}$ has correctly received original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2}+1$ wires during **Phase II** and $L^{\mathbf{R}}_{fault_i}$ is the corresponding L_{fault} sent by $\mathbf{S}$. So $\mathbf{R}$ removes the wires in $L^{\mathbf{R}}_{fault_i}$ from his view for further computation and communication. Now by selecting k_1, k_2 from $\mathcal{K}$ as authentication keys, $\mathbf{R}$ computes $response_2 = URauth(\text{"}continue\text{"}; k_1, k_2)$ where "continue" is a unique pre-defined special signal. $\mathbf{R}$ then send the tuple $(\text{"}continue\text{"}, response_2)$ to $\mathbf{S}$ through the bottom band.

Computation by S at the end of Phase IV:

1. $\mathbf{S}$ checks whether it is getting any 2-tuple identically over at least $\frac{u}{2}+1$ wires. If not, then $\mathbf{S}$ concludes that $\mathbf{R}$ has recovered $m^{\mathbf{R}}$ at the end of **Phase III** and terminates SRMT-Optimal.
2. If $\mathbf{S}$ receives a 2-tuple say $(x^{\mathbf{S}}_1, y^{\mathbf{S}}_1)$ over $\frac{u}{2}+1$ wires, then $\mathbf{S}$ verifies $y^{\mathbf{S}}_1 \stackrel{?}{=} URauth(x^{\mathbf{S}}_1; k_1, k_2)$, where k_1 and k_2 are the keys from $\mathcal{K}$. If the test fails, then $\mathbf{S}$ again concludes that $\mathbf{R}$ has recovered $m^{\mathbf{R}}$ at the end of **Phase III** and terminates SRMT-Optimal.
3. If the test in the previous step succeeds then $\mathbf{S}$ further checks $x^{\mathbf{S}}_1 \stackrel{?}{=}$ "*terminate*". If yes, then $\mathbf{S}$ again concludes that $\mathbf{R}$ has recovered $m^{\mathbf{R}}$ at the end of **Phase III** and terminates SRMT-Optimal. On the other hand, if $x^{\mathbf{S}}_1 \stackrel{?}{=}$ "*continue*" then $\mathbf{S}$ concludes that $\mathcal{Y}^{\mathbf{S}}$ was indeed sent by $\mathbf{R}$ and further continues the protocol.

his consideration for all future computation and communication and will conclude that majority of the wires in the bottom band are honest. This proves the first part of the lemma.

2. $\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$: This case is similar to the case $\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$ in the proof of Lemma 23. Specifically, at the end of **Phase III**, except with probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ will conclude that $\mathbf{S}$ has not correctly received $\mathcal{Y}^{\mathbf{R}}$ over the majority wires in the bottom band during **Phase II**. This further implies that there are at most $t-\frac{u}{2}-1$ corrupted wires in the top band, which further implies that the $F_i^{\mathbf{R}}$'s delivered by the wires in $\mathcal{P}^{\mathbf{R}}$ during **Phase I** are correct, except with error probability $2^{-\Omega(\kappa)}$. So using these $F_i^{\mathbf{R}}$'s, $\mathbf{R}$ will correctly recover $m^{\mathbf{S}}$, except with error probability $2^{-\Omega(\kappa)}$. Moreover, in this case, $\mathbf{R}$ asks $\mathbf{S}$ to terminate the protocol by authenticating "terminate" signal and sending it through the bottom band. Since the keys used for authenticating "terminate" signal are selected from $\mathcal{K}$, they will be unknown to the adversary. So from the properties of $URauth$, the adversary cannot generate authentication of any signal, other than "terminate" and put into the bottom band. So if at all $\mathbf{S}$ receives a valid authenticated signal from at least $\frac{u}{2}+1$ wires in the bottom band, it has to be "terminate" signal, except with error probability $2^{-\Omega(\kappa)}$. On receiving the signal, $\mathbf{S}$ also comes to know that $\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$ and terminates the protocol. On the other hand if $\mathbf{S}$ does not receive a valid authenticated signal from at least $\frac{u}{2}+1$ wires in the bottom band, then also $\mathbf{S}$ comes to know that $\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$ and terminates the protocol. This proves the second part of the lemma.

The fact that $\mathbf{S}$ communicates $\mathcal{O}(n^3)$ field elements during **Phase III** follows from the proof of Lemma 24. During **Phase IV**, $\mathbf{R}$ authenticates either "terminate" and "continue" signal and sends through the entire bottom band. This will require a communication complexity of $\mathcal{O}(u)$ field elements. This proves the third part of the lemma. □

Finally, we move towards the discussion of protocol SRMT-Optimal when $\mathbf{S}$ receives $\mathcal{Y}^{\mathbf{S}}$ over majority wires in the bottom band during **Phase II**, such that the resultant $L_{fault}^{\mathbf{S}}$ is of size more than $t-\frac{u}{2}+1$ and $\mathbf{S}$ has received "continue" signal from $\mathbf{R}$ through the majority wires in the bottom band during **Phase IV**. If this is the case, then from the proof of Lemma 24, except with error probability $2^{-\Omega(\kappa)}$, both $\mathbf{S}$ and $\mathbf{R}$ knows the identity of $|L_{fault}^{\mathbf{S}}| \geq t-\frac{u}{2}+1$ corrupted wires in the top band. This implies that at most $\frac{u}{2}-1$ wires in the bottom band are corrupted and hence majority wires in the bottom band are honest. This further implies that the information received by $\mathbf{R}$ during **Phase I** is insufficient to reconstruct $B^{\mathbf{S}}$. This is because in this case more than $t-\frac{u}{2}+1$ wires have delivered incorrect $F_i^{\mathbf{R}}$'s during **Phase I**. But to reconstruct $B^{\mathbf{S}}$, $\mathbf{R}$ must have the knowledge of $t-\frac{u}{2}+1$ correct $F_i^{\mathbf{R}}$'s.

So $\mathbf{S}$ again starts re-sending $m^{\mathbf{S}}$. Notice that both $\mathbf{S}$ and $\mathbf{R}$ knows that at most $\frac{u}{2}-1$ wires in the top band are corrupted. Moreover, majority of the wires in the bottom band are honest. To resend $m^{\mathbf{S}}$, $\mathbf{S}$ considers only the first $\frac{u}{2}$ wires in the top band. Both $\mathbf{S}$ and $\mathbf{R}$ knows that at least one wire among

these $\frac{u}{2}$ wires are honest. Notice that in order to re-send $m^{\mathbf{S}}$, $\mathbf{S}$ should not communicate more than $\mathcal{O}(|m^{\mathbf{S}}|)$ field elements. This is because the overall goal of SRMT-Optimal is to achieve reliability with constant factor overhead. It seems that there is no way $\mathbf{S}$ can re-send $m^{\mathbf{S}}$ by communicating $\mathcal{O}(|m^{\mathbf{S}}|)$ field elements, as he does not know the identity of the $\frac{u}{2}-1$ corrupted wires in the top band. However, the fact that at least one wire among the first $\frac{u}{2}$ wires in the top band is honest and majority wires in the bottom band are honest comes to our rescue!!

$\mathbf{S}$ divides $m^{\mathbf{S}}$ into $\frac{u}{2}$ blocks and tries to sequentially send each block, one by one. Specifically, $\mathbf{S}$ considers the first block of $m^{\mathbf{S}}$ and sends it only over wire f_1 and waits for the authenticated feedback from $\mathbf{R}$. If f_1 has correctly delivered the block then $\mathbf{S}$ will come to know this and will continue with the second block. Otherwise with very high probability, $\mathbf{S}$ will come to know that f_1 has not delivered the first block correctly. So $\mathbf{S}$ tells about this to $\mathbf{R}$ by sending an authenticated signal and then again send the first block of $m^{\mathbf{S}}$ through the second wire. This process will continue till either all the blocks of $m^{\mathbf{S}}$ have been delivered or $\mathbf{S}$ and $\mathbf{R}$ has tried the first $\frac{u}{2}-1$ wires. In the later case, both $\mathbf{S}$ and $\mathbf{R}$ knows that wire $f_{\frac{u}{2}}$ is honest and so $\mathbf{S}$ sends all the remaining blocks of $m^{\mathbf{S}}$ to $\mathbf{R}$ through wire $f_{\frac{u}{2}}$. It is easy to see that this entire process will take $\mathcal{O}(u)$ phases and will require a communication complexity of $\mathcal{O}(|m^{\mathbf{S}}|)$ field elements. The formal details of the protocol steps are given in Fig. 14.

We now prove the properties of protocol SRMT-Optimal if the protocol follows the steps given in Fig. 14.

Lemma 25. *Suppose* $\mathbf{S}$ *receives the same conflict list* $\mathcal{Y}^{\mathbf{S}}$ *during* **Phase II** *through at least* $\frac{u}{2}+1$ *wires in the bottom band, such that the size of resultant* $L^{\mathbf{S}}_{fault}$ *is more than* $(t-\frac{u}{2}+1)$. *Moreover let* $\mathbf{S}$ *receive "continue" signal during* **Phase IV** *through majority wires in the bottom band. Then except with error probability* $2^{-\Omega(\kappa)}$, $\mathbf{S}$ *will be able to correctly re-send* $m^{\mathbf{S}}$ *to* $\mathbf{R}$ *in* $\mathcal{O}(u)$ *phases by following the steps given in Fig. 14. Moreover, this requires a communication complexity of* $\mathcal{O}(|m^{\mathbf{S}}|) = \mathcal{O}(n^3)$ *field elements.*

PROOF: Suppose $\mathbf{S}$ has received the same conflict list $\mathcal{Y}^{\mathbf{S}}$ during **Phase II** through at least $\frac{u}{2}+1$ wires in the bottom band, such that the size of resultant $L^{\mathbf{S}}_{fault}$ is more than $(t-\frac{u}{2}+1)$. Moreover let $\mathbf{S}$ has received "continue" signal during **Phase IV** through majority wires in the bottom band. Then from the proof of first part of Lemma 24, except with error probability $2^{-\Omega(\kappa)}$, at most $\frac{u}{2}-1$ wires in the top band and at most $\frac{u}{2}-1$ wires in the bottom band are corrupted. Moreover, $\mathbf{S}$ and $\mathbf{R}$ will know this.

In Fig. 14, $\mathbf{S}$ resends $m^{\mathbf{S}}$ by using only the first $\frac{u}{2}$ wires. At a time, $\mathbf{S}$ tries to send only one block of $m^{\mathbf{S}}$. Suppose $\mathbf{S}$ sends the block $B^{\mathbf{S}}_{bc^{\mathbf{S}}}$ to $\mathbf{R}$ over the wire $f_{wc^{\mathbf{S}}}$. If wire $f_{wc^{\mathbf{S}}}$ is honest then the block will be delivered correctly and $\mathbf{S}$ will come to know about this through the feedback paths. $\mathbf{S}$ then asks $\mathbf{R}$ to increment the block count by authenticating a special signal, where the keys for authentication are selected from $\mathcal{K}$ and hence will be unknown to the adversary. So except with error probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ will correctly receive the signal and will increment the block count.

Figure 14: Execution of SRMT-Optimal If **S** Receives "continue" Signal Through the Majority Wires in the Bottom Band at the End of **Phase IV**

1. **S** divides $m^{\mathbf{S}}$ into blocks $B_1^{\mathbf{S}}, \ldots, B_{\frac{u}{2}}^{\mathbf{S}}$, each of size $\frac{|m^{\mathbf{S}}|}{\frac{u}{2}}$ field elements.
2. **S** and **R** initializes variables $wc^{\mathbf{S}} = 1, bc^{\mathbf{S}} = 1$ and $wc^{\mathbf{R}} = 1, bc^{\mathbf{R}} = 1$ respectively. Here wc stands for wire count and bc stands for block count.
3. **S** and **R** now executes the following steps:
   (a) While $(wc^{\mathbf{S}} \leq \frac{u}{2} - 1)$ and (all the blocks of $m^{\mathbf{S}}$ are not delivered to **R**) do the following:
      i. **S** sends the block $B_{bc^{\mathbf{S}}}^{\mathbf{S}}$ to **R** *only* over the wire $f_{wc^{\mathbf{S}}}$ in the top band.
      ii. Let **R** receive $B_{bc^{\mathbf{R}}}^{\mathbf{R}}$ along wire $f_{wc^{\mathbf{R}}}$. Now by selecting a hash key k_{bc} from the set $\mathcal{K}$, **R** computes $x_{bc}^{\mathbf{R}} = hash(k_{bc}; B_{bc^{\mathbf{R}}}^{\mathbf{R}})$ and sends $x_{bc}^{\mathbf{R}}$ to **S** through the entire bottom band.
      iii. **S** correctly receives $x_{bc}^{\mathbf{R}}$ through at least $\frac{u}{2}+1$ wires (recall that in this case majority wires in bottom band are honest) and verifies $x_{bc}^{\mathbf{R}} \stackrel{?}{=} hash(k_{bc}; B_{bc^{\mathbf{S}}}^{\mathbf{S}})$.
      iv. If the test fails then **S** concludes that wire $f_{wc^{\mathbf{S}}}$ has delivered incorrect $B_{bc^{\mathbf{S}}}^{\mathbf{S}}$ to **R**. So **S** does the following:
         A. **S** increments $wc^{\mathbf{S}}$ by one.
         B. **S** authenticates an unique, special, pre-defined signal "increment-wire" by using two keys from the set $\mathcal{K}$ and sends the authenticated signal to **R** through the top band.
         C. **R** correctly receives the signal with very high probability and accordingly increments $wc^{\mathbf{R}}$ by one.
      v. If the test succeeds then **S** concludes that wire $f_{wc^{\mathbf{S}}}$ has delivered correct $B_{bc^{\mathbf{S}}}^{\mathbf{S}}$ to **R**. So **S** does the following:
         A. **S** increments $bc^{\mathbf{S}}$ by one.
         B. **S** authenticates an unique, pre-defined, special "increment-block" signal by using keys from the set $\mathcal{K}$ and sends it to **R** through the top band.
         C. **R** correctly receives the signal with very high probability and accordingly increments $bc^{\mathbf{R}}$ by one.
   (b) If all the blocks of $m^{\mathbf{S}}$ are delivered then both **S** and **R** terminates. Otherwise **S** concatenates all the remaining blocks of $m^{\mathbf{S}}$ and sends it to **R** through wire $f_{\frac{u}{2}}$ and terminates. **R** correctly receives these blocks and terminates.

On the other hand, if $f_{wc^{\mathbf{S}}}$ is corrupted and delivers incorrect block, then except with error probability $2^{-\Omega(\kappa)}$, $\mathbf{S}$ will come to know about this. This is because, $\mathbf{R}$ sends the hash value of the received block where the hash key is selected from $\mathcal{K}$ and hence will be unknown to the adversary (but will be known to $\mathbf{S}$). In this case, $\mathbf{S}$ asks $\mathbf{R}$ to increment the the wire count by authenticating a special signal, where the keys for authentication are selected from $\mathcal{K}$ and hence will be unknown to the adversary. So except with error probability $2^{-\Omega(\kappa)}$, $\mathbf{R}$ will correctly receive the signal and will increment the wire count.

It is now easy to see that it will take $\mathcal{O}(u)$ phases, at the end of which either all the blocks of $m^{\mathbf{S}}$ would be delivered or both $\mathbf{S}$, $\mathbf{R}$ will come to know that first $\frac{u}{2}-1$ wires in the top band are corrupted. In the later case, $\mathbf{S}$ will re-send the remaining blocks of $m^{\mathbf{S}}$ in a single phase through wire $f_{\frac{u}{2}}$, which is bound to be honest. It is easy to see that the overall communication complexity for resending $m^{\mathbf{S}}$ is $\mathcal{O}(|m^{\mathbf{S}}|) = \mathcal{O}(n^3)$ field elements. □

Since there are so many execution sequence possible in protocol SRMT-Optimal, we summarize the steps of protocol SRMT-Optimal in Fig. 15.

We now finally state the following theorem:

Theorem 8. *Suppose there exists $0 \leq u \leq t$ wires in the bottom band and $n = \max(2t-u+1, t+1)$ wires in the top band. Then there exists an $\mathcal{O}(u)$ SRMT protocol which reliably sends a message containing $\ell = (t-\frac{u}{2}+1)n^2 = \Theta(n^3)$ field elements by communicating $\mathcal{O}(\ell) = \mathcal{O}(n^3)$ field elements. In terms of bits, the protocol sends $\Theta(n^3\kappa)$ bits by communicating $\mathcal{O}(n^3\kappa)$ bits. Thus the protocol achieves reliability with constant factor overhead.*

Proof: Follows from the protocol steps summarized in Fig. 15. □

Once we have an SRMT protocol, which achieves reliability with constant factor overhead, we can easily design a communication optimal SSMT protocol, which we do in the next section.

8. An $\mathcal{O}(u)$ Phase Communication Optimal SSMT Protocol

We now design an $\mathcal{O}(u)$ phase SSMT protocol called SSMT-Optimal, which sends a message $m^{\mathbf{S}}$ containing ℓ field elements by communicating $\mathcal{O}(n^3)$ field elements. If the full bottom band is corrupted then $\ell = \Theta(n^2u)$, otherwise $\ell = \Theta(n^2)$. The protocol uses protocols 6-Pad and SRMT-Optimal as black box. The protocol is given in Fig. 16.

We now state the properties of protocol SSMT-Optimal.

Theorem 9. *Protocol SSMT-Optimal is an $\mathcal{O}(u)$ phase SSMT protocol with a communication complexity of $\mathcal{O}(n^3)$ field elements. If the entire bottom band is corrupted then the protocol securely sends $\Theta(n^2u)$ field elements. Otherwise, the protocol securely sends $\Theta(n^2)$ field elements. In terms of bits, the protocol sends either $\Theta(n^2u\kappa)$ or $\Theta(n^3\kappa)$ bits by communicating $\mathcal{O}(n^3\kappa)$ bits, depending upon whether the entire bottom band is corrupted or not.*

Figure 15: Summary of the Steps Executed in Protocol SRMT-Optimal

1. **S** and **R** executes first two phases as shown in Fig. 9. This requires a communication complexity of $\mathcal{O}(n^3)$ field elements (see Claim 24).
2. If a unique conflict list is not received by **S** at the end of **Phase II** through at least $\frac{u}{2}+1$ wires in the bottom band then **S** and **R** executes the steps given in Fig. 10. In this case, protocol terminates at the end of **Phase III** and the overall communication complexity of the protocol is $\mathcal{O}(n^3)$ field elements (see Lemma 22).
3. If a unique conflict list $\mathcal{Y}^{\mathbf{S}}$ is received by **S** at the end of **Phase II** through at least $\frac{u}{2}+1$ wires in the bottom band then **S** computes $L^{\mathbf{S}}_{fault}$ from $\mathcal{Y}^{\mathbf{S}}$ by following the steps given in Fig. 11.
4. If $|L^{\mathbf{S}}_{fault}| \leq (t-\frac{u}{2})$ then **S** and **R** executes the steps given in Fig. 12. In this case, protocol terminates at the end of **Phase III** and the overall communication complexity of the protocol is $\mathcal{O}(n^3)$ field elements (see Lemma 23).
5. If $|L^{\mathbf{S}}_{fault}| \geq (t-\frac{u}{2}+1)$ then **S** and **R** executes the steps given in Fig. 13. Now there are two possible cases:
   (a) If **R** terminates the protocol at the end of **Phase III** then **S** will also terminate the protocol at the end of **Phase IV**. In this case, the communication complexity of the protocol will be $\mathcal{O}(n^3)$ field elements (see Lemma 24).
   (b) If **R** decides to continue the protocol then at the end of **Phase IV**, **S** will also come to know this. In this case, **S** and **R** executes the steps given in Fig. 14. The protocol will require $\mathcal{O}(u)$ phases and a communication complexity of $\mathcal{O}(n^3)$ field elements (see Lemma 25).

Figure 16: An $\mathcal{O}(u)$ Phase Communication Optimal SSMT Protocol SSMT-Optimal

1. **S** and **R** securely establishes a random, non-zero one time pad Pad by executing the six phase protocol 6-Pad. If the entire bottom band is corrupted then the size of the pad is $\Theta(n^2u)$ field elements, otherwise the size of the pad is $\Theta(n^2)$ field elements.
2. If Pad is of size $\Theta(n^2u)$ field elements, then **S** selects a secret message $m^{\mathbf{S}}$ containing $\Theta(n^2u)$ field elements. **S** then computes $C = m^{\mathbf{S}} \oplus Pad$. **S** then appends some extra field elements to C from $\mathbb{F}$, such that C contains $\Theta(n^3)$ field elements. The appended elements are randomly selected from $\mathbb{F}$. Finally, **S** reliably sends C to **R** by executing the protocol SRMT-Optimal (the random elements are appended to C so that C contains $\Theta(n^3)$ field elements. This is because protocol SRMT-Optimal requires the minimum message size to be $\Theta(n^3)$ field elements). **R** correctly receives C with very high probability. **R** then remove the last elements from C, such that C contains $\Theta(n^2u)$ field elements. Finally **R** computes $m^{\mathbf{S}} = C \oplus Pad$ and terminates the protocol.
3. If Pad contains $\Theta(n^2)$ field elements, then **S** and **R** does the same computation as above, except that $m^{\mathbf{S}}$ and original C will be of size $\Theta(n^2)$ field elements.

PROOF: The proof is straightforward and follows from the protocol steps and the properties of protocol 6-Pad and SRMT-Optimal. □

We next show that the communication complexity of protocol SRMT-Optimal and SSMT-Optimal is asymptotically optimal. For this, we derive the lower bound on the communication complexity of SRMT and SSMT protocol in the next section.

9. Lower Bound on Communication Complexity of SRMT and SSMT Protocols

We now derive the lower bound on the communication complexity of SRMT and SSMT protocols. We will then show that our protocols SRMT-Optimal and SSMT-Optimal satisfy these bounds asymptotically. We first begin with the lower bound for SRMT protocols.

9.1. Lower Bound for SRMT Protocols

The lower bound on the communication complexity of SRMT protocols is given by the following theorem:

Theorem 10. *Any SRMT protocol has to communicate $\Omega(\ell)$ field elements to reliably send a message containing ℓ field elements.*

PROOF: The proof simply follows from the fact that any SRMT protocol has to at least communicate the message. □

In the light of the above theorem, we state the following theorem:

Theorem 11. *Protocol SRMT-Optimal is a communication optimal SRMT protocol.*

PROOF: Follows from Theorem 10 and the fact that SRMT-Optimal reliably sends a message containing $\Theta(n^3)$ field elements by communicating $\mathcal{O}(n^3)$ field elements. □

In the next section, we derive the lower bound for SSMT protocols.

9.2. Lower Bound for SSMT Protocols

The derivation of the lower bound on the communication complexity of SSMT protocol is divided into two parts: (a) if the entire bottom band is corrupted and (b) if the entire bottom band is not corrupted. We first consider the case when the entire bottom band is corrupted.

Theorem 12. *Suppose there exists $u \leq t$ wires in the bottom band and $n = 2t - u + 1$ wires in the top band. Moreover, suppose the entire bottom band is corrupted (***S** *and* **R** *may or may not know about this). Then any multiphase SSMT protocol to securely send a message m containing ℓ field elements from $\mathbb{F}$, needs to communicate $\Omega(\frac{n\ell}{u})$ field elements. In terms of bits, the protocol needs to communicate $\Omega(\frac{n\ell}{u}\kappa)$ bits to securely send $\ell\kappa$ bits.*

PROOF: Suppose both **S** and **R** in advance knows that the entire bottom band is corrupted. Under this condition, any multiphase SSMT protocol Π to securely send m, is virtually reduced to a single phase SSMT protocol, where **S** is connected to **R** by $n = 2t - u + 1$ wires, of which at most $t - u$ are corrupted. Since perfect secrecy is required in SSMT, the data sent along the top band in Π must be such that data along any set of $(t - u)$ wires has no information about the secret message m. Otherwise the adversary will also know the secret message by passively listening the contents of these wires. Similarly, the data sent over any set of $(n - (t - u))$ wires over the top band should have full information about the secret message m. The latter requirement ensures that even if the adversary simply blocks all the data that he can, the secret message is not lost and therefore the receiver's ability to recover the message is not completely ruled out. We now define the following notations:

1. $\mathcal{M}$ denotes the message space from where the message m is selected. In our context, $\mathcal{M} = \mathbb{F}^\ell$.
2. For $i = 1, \ldots, n$, $\mathbf{X}_i^m$ denotes the set of all possible transmission in protocol Π, that could occur over wire f_i, corresponding to message $m \in \mathcal{M}$.
3. For $j \geq i$, $\mathbf{M}_{i,j}^m \subseteq \mathbf{X}_i^m \times \mathbf{X}_{i+1}^m \times \ldots \times \mathbf{X}_j^m$ denotes the set of all possible transmission that could occur over wire $f_i, f_{i+1}, \ldots, f_j$ during protocol Π, corresponding to message $m \in \mathcal{M}$.
4. $\mathbf{M}_{i,j} = \bigcup_{m \in \mathcal{M}} \mathbf{M}_{i,j}^m$ and $\mathbf{X}_i = \bigcup_{m \in \mathcal{M}} \mathbf{X}_i^m$. We call $\mathbf{X}_i$ as the *capacity* of wire f_i and $\mathbf{M}_{i,j}$ as the *capacity* of the set of wires $\{f_i, \ldots, f_j\}$.

Now in protocol Π, one element from the set $\mathbf{X}_i$ is transmitted over wire f_i, for $i = 1, \ldots, n$. Moreover, each element of the set $\mathbf{X}_i$ can be represented by $\log |\mathbf{X}_i|$ bits. Thus, if we can find out each $\mathbf{X}_i$, then the lower bound on the communication complexity of Π will be $\Sigma_{i=1}^n \log |\mathbf{X}_i|$ bits. In the sequel, we try to estimate $\mathbf{X}_i$.

From the properties of data sent over the top band in protocol Π, the data sent over any set of $t - u$ wires is *independent* of the message. Thus, for any two messages $m_1, m_2 \in \mathcal{M}$, it must hold that

$$\mathbf{M}_{t-u+1,2(t-u)}^{m_1} = \mathbf{M}_{t-u+1,2(t-u)}^{m_2}.$$

Notice that the above relation must hold for any selection of $t-u$ wires in the top band. We focussed on the set of wires $\{f_{t-u+1}, \ldots, f_{2(t-u)}\}$ just for simplicity.

Also, from the properties of data transmitted over the top band in Π, the data transmitted over any set of $n - (t - u)$ wires should have *full* information

about the message m and hence uniquely determine m. Thus it must also hold that

$$\mathbf{M}^{m_1}_{t-u+1,n} \cap \mathbf{M}^{m_2}_{t-u+1,n} = \emptyset.$$

We again stress that the above relation must hold for any set of $n-(t-u)$ wires in the top band. We focussed on the set of wires $\{f_{t-u+1}, \ldots, f_n\}$ just for simplicity.

As mentioned earlier, $\mathbf{M}^m_{t-u+1,2(t-u)}$ will be same for all messages m. Thus, in order that $\mathbf{M}^{m_1}_{t-u+1,n} \cap \mathbf{M}^{m_2}_{t-u+1,n} = \emptyset$ holds, it must be the case that $\mathbf{M}^m_{2(t-u)+1,n}$ is *unique* for every message m. This implies that

$$|\mathbf{M}_{2(t-u)+1,n}| = |\mathcal{M}|.$$

From the definition of $\mathbf{X}_i$ and $\mathbf{M}_{i,j}$, we get

$$\Pi^n_{i=2(t-u)+1}|\mathbf{X}_i| \geq |\mathbf{M}_{2(t-u)+1,n}| \geq |\mathcal{M}|.$$

Let $g = n - 2(t-u)$. The above inequality holds for any set of g wires $\mathcal{D}$ in the top band, where $|\mathcal{D}| = g$; i.e., $\Pi_{f_i \in \mathcal{D}}|\mathbf{X}_i| \geq |\mathcal{M}|$. In particular, it holds for every selection $\mathcal{D}_k$ of set of wires $\{f_{(kg+1) \bmod n}, f_{(kg+2) \bmod n}, \ldots, f_{(kg+g) \bmod n}\}$, with $k \in \{0, \ldots, n-1\}$.

If we consider all above $\mathcal{D}_k$ sets, then each wire is counted for exactly g times. Thus, the product of the capacities of all $\mathcal{D}_k$ yields the capacity of the full top band to the g-th power, and since each $\mathcal{D}_k$ has capacity at least $|\mathcal{M}|$, we get

$$|\mathcal{M}|^n \leq \Pi^{n-1}_{k=0}\Pi_{f_j \in \mathcal{D}_k}|\mathbf{X}_j| = \left(\Pi^n_{i=1}|\mathbf{X}_i|\right)^g,$$

and therefore

$$n\log(|\mathcal{M}|) \leq g\Sigma^n_{i=1}\log(|\mathbf{X}_i|).$$

As $\log(|\mathcal{M}|) = \ell\log(|\mathbb{F}|)$, from the above inequality, we get

$$\Sigma^n_{i=1}\log(|\mathbf{X}_i|) \geq \left(\frac{n\ell\log(|\mathbb{F}|)}{g}\right) \geq \left(\frac{n\ell\log(|\mathbb{F}|)}{n-2(t-u)}\right).$$

As mentioned earlier, $\Sigma^n_{i=1}\log(|\mathbf{X}_i|)$ denotes the lower bound on the communication complexity of protocol Π. From the above inequality, we find that the lower bound on the communication complexity of protocol Π is $\Omega\left(\frac{n\ell\log(|\mathbb{F}|)}{n-2(t-u)}\right) = \Omega\left(\frac{n\ell}{n-2(t-u)}\kappa\right)$ bits. Now each field element from $\mathbb{F}$ can be preprsented by κ bits. Thus the lower bound on the communication complexity of protocol Π is $\Omega\left(\frac{n\ell}{n-2(t-u)}\right) = \Omega(\frac{n\ell}{u})$ field elements, as $n = 2t-u+1$. □

We now proceed to the second case when the entire bottom band is not corrupted. The lower bound on the communication complexity of SSMT protocol for this case is given by the following theorem:

Theorem 13. *Suppose there exists $u \leq t$ wires in the bottom band and $n = 2t - u + 1$ wires in the top band. Moreover, suppose that the entire bottom band is not corrupted. Then any multiphase SSMT protocol to securely send a message m containing ℓ field elements from $\mathbb{F}$, needs to communicate $\Omega(n\ell)$ field elements. In terms of bits, the protocol needs to communicate $\Omega(n\ell\kappa)$ bits to securely send $\ell\kappa$ bits.*

PROOF: Suppose there exists $u \leq t$ wires in the bottom band and $n = 2t-u+1$ wires in the top band. Let $N = (n+u) = 2t+1$. Now $n = \Theta(t)$, as there exists at least $t+1$ wires in the top band. Also $N = \Theta(t)$. Let Π be a multiphase SSMT protocol over the N wires to securely transmit a message m containing ℓ field elements from $\mathbb{F}$. Then in any execution of Π, the data exchanged along any set of t wires (including top and bottom band) must be independent of m. Otherwise, adversary can passively listen these wires and will know m, which violates perfect secrecy condition of Π. On the other hand, data exchanged along any set of $N - t$ wires (including top and bottom band) should have complete information about the message. The latter requirement ensures that even if the adversary simply blocks all the data that he can, the secret message is not lost and therefore the receiver's ability to recover the message is not completely ruled out. Let the N wires between $\mathbf{S}$ and $\mathbf{R}$ be denoted by $w_1, \ldots, w_N$. Out of these N wires, the first n wires are the wires from the top band, which are directed from $\mathbf{S}$ to $\mathbf{R}$. On the other hand, remaining $N - n = u$ wires are from the bottom band, which are directed from $\mathbf{R}$ to $\mathbf{S}$.

We now define the following notations:

1. $\mathcal{M}$ denotes the message space from where the message m is selected. In our context, $\mathcal{M} = \mathbb{F}^\ell$.
2. For $i = 1, \ldots, N$, $\mathbf{X}_i^m$ denotes the set of all possible transmission in protocol Π, that could occur over wire w_i, corresponding to message $m \in \mathcal{M}$.
3. For $j \geq i$, $\mathbf{M}_{i,j}^m \subseteq \mathbf{X}_i^m \times \mathbf{X}_{i+1}^m \times \ldots \times \mathbf{X}_j^m$ denotes the set of all possible transmission that could occur over wire $w_i, w_{i+1}, \ldots, w_j$ during protocol Π, corresponding to message $m \in \mathcal{M}$.
4. $\mathbf{M}_{i,j} = \bigcup_{m \in \mathcal{M}} \mathbf{M}_{i,j}^m$ and $\mathbf{X}_i = \bigcup_{m \in \mathcal{M}} \mathbf{X}_i^m$. We call $\mathbf{X}_i$ as the *capacity* of wire w_i and $\mathbf{M}_{i,j}$ as the *capacity* of the set of wires $\{w_i, \ldots, w_j\}$.

Now in protocol Π, one element from the set $\mathbf{X}_i$ is transmitted over wire w_i, for $i = 1, \ldots, N$. Moreover, each element of the set $\mathbf{X}_i$ can be represented by $\log |\mathbf{X}_i|$ bits. Thus, if we can find out each $\mathbf{X}_i$, then the lower bound on the communication complexity of Π will be $\Sigma_{i=1}^N \log |\mathbf{X}_i|$ bits. In the sequel, we try to estimate $\mathbf{X}_i$.

From the properties of protocol Π, the data sent over any set of t wires is *independent* of the message. Thus, for any two messages $m_1, m_2 \in \mathcal{M}$, it must hold that

$$\mathbf{M}_{t+1,2t}^{m_1} = \mathbf{M}_{t+1,2t}^{m_2}.$$

Notice that the relation above must hold for any selection of t wires out of the N wires between $\mathbf{S}$ and $\mathbf{R}$. We focussed on the set of wires $\{w_{t+1}, \ldots, w_{2t}\}$ just for simplicity.

Also, from the properties of protocol Π, the data transmitted over any set of $N-t$ wires should have *full* information about the message m and hence uniquely determine m. Thus it must also hold that

$$\mathbf{M}_{t+1,N}^{m_1} \cap \mathbf{M}_{t+1,N}^{m_2} = \emptyset.$$

We again stress that the above relation must hold for any set of $N-t$ wires out of the N wires between $\mathbf{S}$ and $\mathbf{R}$. We focussed on the set of wires $\{w_{t+1}, \ldots, w_N\}$ just for simplicity.

As mentioned earlier, $\mathbf{M}_{t+1,2t}^{m}$ will be same for all messages m. Thus, in order that $\mathbf{M}_{t+1,N}^{m_1} \cap \mathbf{M}_{t+1,N}^{m_2} = \emptyset$ holds, it must be the case that $\mathbf{M}_{2t+1,N}^{m}$ is *unique* for every message m. This implies that

$$|\mathbf{M}_{2t+1,N}| = |\mathcal{M}|.$$

From the definition of $\mathbf{X}_i$ and $\mathbf{M}_{i,j}$, we get

$$\Pi_{i=2t+1}^{N} |\mathbf{X}_i| \geq |\mathbf{M}_{2t+1,N}| \geq |\mathcal{M}|.$$

Let $g = N - 2t$. The above inequality holds for any set of g wires $\mathcal{D}$, where $|\mathcal{D}| = g$; i.e., $\Pi_{w_i \in \mathcal{D}} |\mathbf{X}_i| \geq |\mathcal{M}|$. In particular, it holds for every selection $\mathcal{D}_k$ of set of wires $\{w_{(kg+1) \bmod N}, w_{(kg+2) \bmod N}, \ldots, w_{(kg+g) \bmod N}\}$, with $k \in \{0, \ldots, N-1\}$.

If we consider all above $\mathcal{D}_k$ sets, then each wire is counted exactly g times. Thus, the product of the capacities of all $\mathcal{D}_k$ yields the capacity of the full top band and bottom band to the g-th power, and since each $\mathcal{D}_k$ has capacity at least $|\mathcal{M}|$, we get

$$|\mathcal{M}|^N \leq \Pi_{k=0}^{N-1} \Pi_{w_j \in \mathcal{D}_k} |\mathbf{X}_j| = \left(\Pi_{i=1}^{N} |\mathbf{X}_i|\right)^g,$$

and therefore

$$N \log(|\mathcal{M}|) \leq g \Sigma_{i=1}^{N} \log(|\mathbf{X}_i|).$$

As $\log(|\mathcal{M}|) = \ell \log(|\mathbb{F}|)$, from the above inequality, we get

$$\Sigma_{i=1}^{N} \log(|\mathbf{X}_i|) \geq \left(\frac{N\ell \log(|\mathbb{F}|)}{g}\right) \geq \left(\frac{N\ell \log(|\mathbb{F}|)}{N-2t}\right).$$

As mentioned earlier, $\Sigma_{i=1}^{N} \log(|\mathbf{X}_i|)$ denotes the lower bound on the communication complexity of protocol Π. From the above inequality, we find that the lower bound on the communication complexity of protocol Π is $\Omega\left(\frac{N\ell \log(|\mathbb{F}|)}{N-2t}\right) = \Omega\left(N\ell\kappa\right)$ bits, as $N = 2t+1$. Now each field element from $\mathbb{F}$ can be preprsented by κ bits. Thus the lower bound on the communication complexity of protocol Π is $\Omega(N\ell) = \Omega(n\ell)$ field elements, as $N = \Theta(t) = \Theta(n)$. □

Now in the light of previous two theorems, we can state the following theorem:

Theorem 14. *Protocol SSMT-Optimal is a communication optimal SSMT protocol.*

PROOF: From Theorem 9, if the entire bottom band is corrupted then protocol SSMT-Optimal securely sends $\ell = \Theta(n^2u)$ field elements by communicating $\mathcal{O}(n^3)$ field elements. From Theorem 12, if the entire bottom band is corrupted, then any multiphase SSMT protocol has to communicate $\Omega(\frac{n\ell}{u}) = \Omega(n^3)$ field elements to securely send $\ell = \Theta(n^2u)$ field elements. So if the entire bottom band is corrupted then protocol SSMT-Optimal is a communication optimal SSMT protocol.

If the entire bottom band is not corrupted then protocol SSMT-Optimal securely sends $\ell = \Theta(n^2)$ field elements by communicating $\mathcal{O}(n^3)$ field elements. From Theorem 13, if the entire bottom band is not corrupted, then any multiphase SSMT protocol has to communicate $\Omega(n\ell) = \Omega(n^3)$ field elements to securely send $\ell = \Theta(n^2)$ field elements. So if the entire bottom band is not corrupted then also protocol SSMT-Optimal is a communication optimal SSMT protocol. □

10. Conclusion

In this paper, we have resolved the issues related to the FEASIBILITY and OPTIMALITY of SRMT and SSMT protocols in directed networks, which is done for the first time in the literature of RMT and SMT protocols. This paper leaves certain open problems. Our communication optimal SRMT and SSMT protocol require $\mathcal{O}(u)$ phases. It would be interesting to come up with communication optimal SRMT and SSMT protocols with less phase complexity. Our SRMT and SSMT protocol is communication optimal only for messages of some minimum specific length. It would be interesting to design SRMT and SSMT protocols which are communication optimal for messages of any length.

References

[1] Clique (graph theory). Wikipedia article.

[2] Maximal independent set. Wikipedia article.

[3] Tuŕan graph. Wikipedia article.

[4] S. Agarwal, R. Cramer, and R. de Haan. Asymptotically Optimal Two-Round Perfectly Secure Message Transmission. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 394–408. Springer-Verlag, 2006.

[5] B. V. Ashwinkumar, A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. On tradeoff between network connectivity, phase complexity and communication complexity of reliable communication tolerarting mixed adversary. In R. A. Bazzi and B. Patt-Shamir, editors, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed*

Computing, PODC 2008, Toronto, Canada, August 18-21, 2008, pages 115–124. ACM, 2008.

[6] Z. Beerliová-Trubíniová and M. Hirt. Efficient Multi-party Computation with Dispute Control. In S. Halevi and T. Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 305–328. Springer, 2006.

[7] Z. Beerliová-Trubíniová and M. Hirt. Perfectly-Secure MPC with Linear Communication Complexity. In R. Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 213–230. Springer, 2008.

[8] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, 2-4 May 1988, Chicago, Illinois, USA*, pages 1–10. ACM, 1988.

[9] R. Canetti and T. Rabin. Fast Asynchronous Byzantine Agreement with Optimal Resilience. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, 1993*, pages 42–51. ACM, 1993.

[10] D. Chaum, C. Crépeau, and I. Damgård. Multiparty Unconditionally Secure Protocols (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, 2-4 May 1988, Chicago, Illinois, USA*, pages 11–19. ACM, 1988.

[11] A. Choudhary, A. Patra, B. V. Ashwinkumar, K. Srinathan, and C. Pandu Rangan. Perfectly Reliable and Secure Communication Tolerating Static and Mobile Mixed Adversary. In R. Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 137–155. Springer, 2008.

[12] A. Choudhary, A. Patra, Ashwinkumar B. V, K. Srinathan, and C. Pandu Rangan. On minimal connectivity requirement for secure message transmission in asynchronous networks. In V. Garg, R. Wattenhofer, and K. Kothapalli, editors, *Distributed Computing and Networking, 10th International Conference, ICDCN 2009, Hyderabad, India, January 03-06, 2009*, volume 5408 of *Lecture Notes in Computer Science*, pages 148–162, 2009.

[13] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient Multiparty Computations Secure Against an Adaptive Adversary. In J. Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 311–326. Springer, 1999.

[14] Y. Desmedt and Y. Wang. Perfectly Secure Message Transmission Revisited. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 502–517. Springer, 2003.

[15] Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In *Algorithms and Computation, 16th International Symposium, ISAAC 2005, Sanya, Hainan, China, December 19-21, 2005, Proceedings*, volume 3827 of *Lecture Notes in Computer Science*, pages 277–287. Springer Verlag, 2005.

[16] D. Dolev. The Byzantine Generals Strike Again. *Journal of Algorithms*, 3:14–30, 1982.

[17] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly Secure Message Transmission. *JACM*, 40(1):17–47, 1993.

[18] P. Feldman and S. Micali. An optimal algorithm for synchronous Byzantine Agreemet. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 639–648. ACM Press, 1988.

[19] P. Feldman and S. Micali. An optimal probabilistic algorithm for synchronous Byzantine Agreement. In G. Ausiello, M. Dezani-Ciancaglini, and S. R. D. Rocca, editors, *Automata, Languages and Programming, 16th International Colloquium, ICALP89, Stresa, Italy, July 11-15, 1989, Proceedings*, volume 372 of *Lecture Notes in Computer Science*, pages 341–378. Springer Verlag, 1989.

[20] M. Fitzi, M. K. Franklin, J. A. Garay, and S. Harsha Vardhan. Towards Optimal and Efficient Perfectly Secure Message Transmission. In S. P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 311–322. Springer, 2007.

[21] M. Franklin and R. Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, 13(1):9–30, 2000.

[22] M. Franklin and M. Yung. Secure Hypergraphs: Privacy from partial broadcast. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 36–44, 1995.

[23] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, USA*, pages 218–229. ACM, 1987.

[24] V. Hadzilacos. *Issues of Fault Tolerance in Concurrent Computations*. PhD thesis, Harvard University, Cambridge, Massachusetts, 1984.

[25] M. V. N. A. Kumar, P. R. Goundan, K. Srinathan, and C. Pandu Rangan. On perfectly secure communication over arbitrary networks. In *PODC 2002, Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing, July 21-24, 2002 Monterey, California, USA*, pages 193–202. ACM, 2002.

[26] K. Kurosawa and K. Suzuki. Truly Efficient 2-Round Perfectly Secure Message Transmission Scheme. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 324–340. Springer, 2008.

[27] K. Kurosawa and K. Suzuki. Almost secure (1-round, n-channel) message transmission scheme. *IEICE Transactions*, 92-A(1):105–112, 2009.

[28] L. Lamport. The weak Byzantine Generals problem. *J. ACM*, 30(3):668–676, 1983.

[29] L. Lamport, R. E. Shostak, and M. C. Pease. The Byzantine Generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[30] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[31] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28, 1965.

[32] A. Narayanan, K. Srinathan, and C. Pandu Rangan. Perfectly Reliable Message Transmission. *Information Processing Letters*, 11(46):1–6, 2006.

[33] A. Patra, A. Choudhary, and C. Pandu Rangan. Constant phase efficient protocols for secure message transmission in directed networks. In I. Gupta and R. Wattenhofer, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 2007, Portland, Oregon, USA, August 12-15, 2007*, pages 322–323. ACM, 2007.

[34] A. Patra, A. Choudhary, and C. Pandu Rangan. Unconditionally reliable and secure message transmission in directed networks revisited. In R. Ostrovsky, R. De Prisco, and I. Visconti, editors, *Security and Cryptography for Networks, 6th International Conference, SCN 2008, Amalfi, Italy, September 10-12, 2008. Proceedings*, volume 5229 of *Lecture Notes in Computer Science*, pages 309–326. Springer, 2008.

[35] A. Patra, A. Choudhary, and C. Pandu Rangan. Brief announcement: Perfectly secure message transmission in directed networks revisited. In S. Tirthapura and L. Alvisi, editors, *Proceedings of the 28th Annual ACM Symposium on Principles of Distributed Computing, PODC 2009, Calgary, Alberta, Canada, August 10-12, 2009*, pages 305–328. ACM, 2009.

[36] A. Patra, A. Choudhary, and C. Pandu Rangan. On communication complexity of secure message transmission in directed networks. In K. Kant, S. V. Premmaraju, K. M. Sivalingam, and J. Wu, editors, *Distributed Computing and Networking, 11th International Conference, ICDCN 2010, Kolkata, India, January 3 - 6, 2010, Proceedings*, volume 5935 of *Lecture Notes in Computer Science*, pages 42–53. Springer Verlag, 2010.

[37] A. Patra, A. Choudhary, C. Pandu Rangan, K. Srinathan, and P. Raghavendra. Perfectly Reliable and Secure Message Transmission Tolerating Mobile Adversary. *International Journal of Applied Cryptography*, 1(3):200 – 224, 2009.

[38] A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Constant phase bit optimal protocols for perfectly reliable and secure message transmission. In R. Barua and T. Lange, editors, *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, volume 4329 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2006.

[39] A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Perfectly reliable and secure communication in directed networks tolerating mixed adversary. In A. Pelc, editor, *Distributed Computing, 21st International Symposium, DISC 2007, Lemesos, Cyprus, September 24-26, 2007, Proceedings*, volume 4731 of *Lecture Notes in Computer Science*, pages 496–498. Springer, 2007.

[40] A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Unconditionally reliable and secure message transmission in undirected synchronous networks: Possibility, feasibility and optimality. Accepted for publication in IJACT, 2009.

[41] A. Patra, B. Shankar, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In F. Bao, S. Ling, T. Okamoto, H. Wang, and C. Xing, editors, *Cryptology and Network Security, 6th International Conference, CANS 2007, Singapore, December 8-10, 2007, Proceedings*, volume 4856 of *Lecture Notes in Computer Science*, pages 80–101. Springer, 2007.

[42] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85. ACM, 1989.

[43] J. Renault and T. Tomala. Probabilistic reliability and privacy of communication using multicast in general neighbor networks. *J. Cryptology*, 21(2):250–279, 2008.

[44] H. Sayeed and H. Abu-Amara. Perfectly Secure Message Transmission in Asynchronous Networks. In *Proceedings of 7th IEEE Symposium on Parallel and Distributed Processing*, pages 100–105. IEEE, 1995.

[45] H. Sayeed and H. Abu-Amara. Efficient perfectly secure message transmission in synchronous networks. *Information and Computation*, 126(1):53–61, 1996.

[46] B. Shanker, P. Gopal, K. Srinathan, and C. Pandu Rangan. Unconditional reliable message transmision in directed networks. In S. Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 1048–1055. SIAM, 2008.

[47] P.W. Shor. Polynomial time algorithms for Prime factorization and Discrete Logarithms on a Quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[48] K. Srinathan, M. V. N. Ashwin Kumar, and C. Pandu Rangan. Asynchronous secure communication tolerating mixed adversaries. In Y. Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 224–242. Springer, 2002.

[49] K. Srinathan, A. Narayanan, and C. Pandu Rangan. Optimal perfectly secure message transmission. In M. K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer, 2004.

[50] K. Srinathan, A. Patra, A. Choudhary, and C. Pandu Rangan. Unconditionally reliable message transmission in directed hypergraphs. In M. K. Franklin, L. C. K. Hui, and D. S. Wong, editors, *Cryptology and Network Security, 7th International Conference, CANS 2008, Hong-Kong, China, December 2-4, 2008. Proceedings*, volume 5339 of *Lecture Notes in Computer Science*, pages 285–303. Springer, 2008.

[51] K. Srinathan, A. Patra, A. Choudhary, and C. Pandu Rangan. Unconditionally secure message transmission in arbitrary directed synchronous networks tolerating generalized mixed adversary. In W. Li, W. Susilo, U. K. Tupakula, R. Safavi-Naini, and V. Varadharajan, editors, *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10-12, 2009*, pages 171–182. ACM, 2009.

[52] K. Srinathan, N. R. Prasad, and C. Pandu Rangan. On the optimal communication complexity of multiphase protocols for perfect communication.

In *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, pages 311–320. IEEE Computer Society, 2007.

[53] K. Srinathan, P. Raghavendra, and C. Pandu Rangan. On proactive perfectly secure message transmission. In J. Pieprzyk, H. Ghodosi, and E. Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 461–473. Springer, 2007.

[54] K. Srinathan and C. Pandu Rangan. Possibility and complexity of probabilistic reliable communication in directed networks. In E. Ruppert and D. Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 265–274. ACM Press, 2006.

[55] Y. Wang and Y. Desmedt. Perfectly secure message transmission revisited. *IEEE Transactions on Information Theory*, 54(6), 2008.

[56] Q. Yang and Y. Desmedt. Cryptanalysis of secure message transmission protocols with feedback. To appear in Proc. of ICITS, 2009.

[57] A. C. Yao. Protocols for secure computations. In *Proceedings of 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, 3-5 November 1982*, pages 160–164. IEEE Computer Scciety, 1982.